



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

UNIVERZÁLNÍ PALUBNÍ POČÍTAČ DO AUTA

UNIVERSAL ONBOARD AUTOMOTIVE COMPUTER

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. PETR NOVOHRADSKÝ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. VÁCLAV ŠIMEK,

BRNO 2015

Abstrakt

Práce obsahuje kompletní návrh prototypu vestavěného zařízení pro zobrazování informací z vozidla. Je popsán čip ELM327 a jeho základní použití. Popisuje také vývoj diagnostické zásuvky vozidel v Evropských zemích. Obsahuje kompletní popis návrhu schématu, desky plošných spojů a popis implementace firmware. Práce popisuje využití realtime operačního systému ve vestavěném zařízení.

Abstract

This work includes the complete design of a prototype embedded system for displaying information from a vehicle. ELM327 chip is described and its basic use. Work also describes the development of the diagnostic socket for vehicles in European countries. Also contains a complete description of developing schematic, printed circuit board and firmware implementation. The thesis describes the use of realtime operating system in embedded system.

Klíčová slova

Automobil, vozidlo, diagnostika, OBD, palubní počítač, ELM327, schéma, PCB, RTOS, LCD.

Keywords

Car, vehicle, dignostics, OBD, on-board computer, ELM327, schematic, PCB, RTOS, LCD.

Citace

Petr Novohradský: Univerzální palubní počítač do auta, diplomová práce, Brno, FIT VUT v Brně, 2015

Univerzální palubní počítač do auta

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Václava Šimka. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Petr Novohradský
27.5.2015

Poděkování

Děkuji vedoucímu práce za konzultace a spolupráci při realizaci a ladění výsledného zařízení.

© Petr Novohradský, 2015.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

| | |
|---|-----------|
| 1 Úvod | 3 |
| 2 Elektronická diagnostika vozidel | 4 |
| 2.1 Standardizace | 4 |
| 2.1.1 Zásuvka diagnostiky | 5 |
| 2.1.2 Formát komunikace | 6 |
| 2.1.3 Kódování chyb | 6 |
| 3 Návrh | 8 |
| 3.1 Koncepce zařízení | 8 |
| 3.1.1 Napájení | 8 |
| 3.1.2 ELM327 | 9 |
| 3.1.3 Displej | 9 |
| 3.1.4 Procesor | 10 |
| 3.1.5 RTC | 11 |
| 3.1.6 Ovládání | 11 |
| 4 Obvod ELM327 | 12 |
| 4.1 Princip funkce a zapojení | 12 |
| 4.1.1 Doporučené zapojení | 13 |
| 4.2 Komunikace s obvodem | 13 |
| 4.2.1 Interní příkazy | 15 |
| 4.2.2 Zaslání OBD příkazů | 15 |
| 4.2.3 Pokročilá komunikace | 16 |
| 5 Návrh schématu | 17 |
| 5.1 Připojení vozidla | 17 |
| 5.1.1 Napájení | 18 |
| 5.1.2 Sběrnice CAN | 18 |
| 5.1.3 Sběrnice ISO 9141-2 a ISO 14230 | 18 |
| 5.1.4 Sběrnice J1850 | 19 |
| 5.1.5 Obvod ELM327 | 19 |
| 5.2 Zapojení periférií | 20 |
| 5.2.1 RTC obvod | 20 |
| 5.2.2 Displej | 20 |
| 5.3 Zapojení procesoru | 21 |
| 5.3.1 Napájení a zdroj hodinového signálu | 21 |
| 5.3.2 Ovládací vstupy | 22 |

| | | |
|----------|---|-----------|
| 5.3.3 | Rozhraní vozidla a periférií | 22 |
| 5.3.4 | Externí paměť | 23 |
| 5.3.5 | Servisní obvody | 23 |
| 5.4 | Ovládací modul na volant | 24 |
| 6 | Návrh tištěného spoje | 25 |
| 6.1 | Příprava návrhu plošného spoje | 25 |
| 6.1.1 | Výběr pouzder součástek | 25 |
| 6.1.2 | Výběr počtu vrstev spoje | 26 |
| 6.1.3 | Výběr výrobní technologie | 26 |
| 6.2 | Návrh desky plošného spoje | 27 |
| 6.2.1 | Pravidla pro umístění součástek | 27 |
| 6.2.2 | Vedení spojů | 28 |
| 6.2.3 | Umístění servisního potisku | 29 |
| 6.2.4 | Data pro výrobu desky plošných spojů | 29 |
| 7 | Firmware zařízení | 30 |
| 7.1 | Výběr programovacího jazyka | 30 |
| 7.2 | Ovládací modul | 30 |
| 7.3 | Operační systém FreeRTOS | 31 |
| 7.3.1 | Přepnutí kontextu | 32 |
| 7.3.2 | Úprava pro konkrétní procesor | 32 |
| 7.3.3 | Nastavení systému | 33 |
| 7.3.4 | Základní použití | 34 |
| 7.4 | Firmware pro hlavní desku | 34 |
| 7.4.1 | Snímání stavu tlačítek | 35 |
| 7.4.2 | Příjem hlášení od ovládacího modulu | 36 |
| 7.4.3 | Komunikace s displejem a uživatelské rozhraní | 36 |
| 7.4.4 | Získávání dat z vozidla | 38 |
| 7.4.5 | Odeslání příkazu pro čip ELM327 | 39 |
| 7.4.6 | Příjem odpovědi na zaslaný příkaz | 39 |
| 8 | Závěr | 40 |
| A | Schéma obvodu | 42 |

Kapitola 1

Úvod

Přestože vývoj v oblasti silničních vozidel vede k větší automatizaci řízení, patří vozidla stále ke strojům řízeným člověkem. Pokud však má člověk ovládat nějaký stroj, musí nejprve vědět, jak se stroj právě chová. Právě k tomu jsou určena nejrůznější měřidla. V případě automobilu patří mezi nejzákladnější provozní údaje aktuální rychlost jízdy a hladina paliva v nádrži. Často se údaje doplňují o další diagnostické a provozní údaje, jako je teplota nebo otáčky motoru.

Moderní automobily měří mnohem větší množství údajů. Naprostá většina z nich však zůstává před řidičem skryta. Naměřené hodnoty se využívají v nejrůznějších elektronických systémech vozidla, ale některé by mohl s výhodou využít i řidič. Například údaje o aktuální spotřebě paliva může řidiče upozornit, že jeho styl jízdy není příliš ekonomický ani ekologický. Údaje také lze využít pro diagnostiku vozidla. V případě neobvyklých hodnot z nich lze usuzovat na technické problémy a provést opravu dříve, než porucha způsobí mnohem větší škody. Někdy je vhodné naměřené údaje dále zpracovat. Příkladem může být údaj o dojezdové vzdálenosti bez tankování, který se vypočítá pomocí údajů o průměrné spotřebě a aktuální hladině paliva v nádrži.

A tak vznikly palubní počítače. Jedná se o elektronické zařízení ve vozidle, které má přístup k měřeným hodnotám. Naměřené hodnoty dále zpracuje a výsledky zobrazuje za jízdy řidiči. Takové palubní počítače mají v nabídce takřka všichni větší výrobci automobilů. Obvykle je to jednoduché zařízení se segmentovým displejem pro zobrazování údajů. Bohužel, obvykle nejsou v základní výbavě vozu. Podobné zařízení s lepším zobrazením hodnot a použitelností na různá vozidla je cílem této práce.

Cílem práce je navrhnout elektronické zařízení pro získávání provozních údajů z vozidla, jejich vhodné zpracování, zobrazení a záznam. Zařízení by mělo být co nejuniverzálnější, použitelné na vozidlech od různých výrobců. Zaměří se tedy na využití standardizovaných komunikačních protokolů ve vozidlech. Práce si neklade za cíl vývoj snímačů. Budou se využívat výhradně údaje, které dané vozidlo poskytne pomocí elektronické komunikace.

Kapitola 2

Elektronická diagnostika vozidel

Protože je ve vozidlech stále více elektronických systémů, přichází s tím také větší možnosti jak provést diagnostiku vozidel. K odhalení problémů lze, kromě klasických mechanických prohlídek, využít záznamy z elektroniky vozidla. Tyto údaje přivedou technika k nalezení problému mnohdy rychleji než klasická prohlídka. Nicméně k přečtení záznamů z elektroniky je také potřeba, aby výrobce připravil možnost, jak takové údaje získat.

Už od prvních elektronických systémů ve vozidlech výrobci mysleli na možnosti diagnostiky takových systémů. Obvykle se jednotka musela vyjmout z vozidla a připojit na diagnostický přístroj. Často se stávalo, že protokol komunikace nebyl zveřejněný, k diagnostice tak bylo třeba použít výhradně zařízení dodávané výrobcem, případně celou jednotku výrobcem zaslat.

2.1 Standardizace

Bylo jasné, že je třeba komunikační protokoly standardizovat. A to nejlépe ve všech vozidlech. Proto v roce 1998 vznikla direktiva Evropské unie s označením *Directive 98/69/EC of the European Parliament* [13]. Toto nařízení se věnuje měření škodlivých emisí motorových vozidel a kromě jiných věcí zavádí jednotný konektor pro diagnostiku. Také specifikuje komunikační protokoly v takovém konektoru.

V důsledku zmíněného nařízení musí být všechny nové osobní vozy v Evropské unii vybavené elektronickou diagnostikou systému pro kontrolu spalín. Také text výslovně zmiňuje, že přístup k informacím z diagnostiky spalín musí být ve standardizovaném formátu a výrobce vozidla jej nesmí nijak omezovat. Zásuvka diagnostiky a její zapojení musí odpovídat normě ISO 15031-3[7]. V zásuvce musí být k dispozici alespoň jedno rozhraní z celkem 3 možných. Systém musí být schopen zaznamenat a pro pozdější analýzu uložit nejméně jednu poslední chybu. Kódy hlášených chyb pak musí odpovídat normě ISO 15031-6[8].

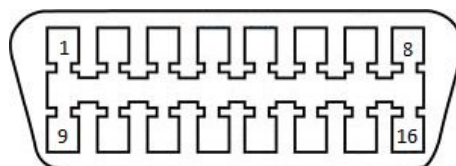
Bohužel důležitým faktem zůstává, že nařízení Evropské unie a zmíněné normy vše řeší pouze z pohledu emisí vozidla. Proto normy vyžadují pouze komunikaci s emisními systémy. Komunikace s ostatními jednotkami vozidla je ponechána na libovůli výrobce a nemusí být vůbec možná. Lze tak například získat informace o nefunkční Lambda sondě ve výfukovém potrubí, ale nelze získat informace třeba o nefunkční navigaci nebo klimatizaci.

Zmíněné normy ve svém důsledku znamenají, že na trhu je dostupné velké množství přístrojů, které chybové kódy z automobilů umí číst. Využívají se především na stanicích technické kontroly. Umožní velice rychle zjistit, zda řídící jednotka nezaznamenala za jízdy nějaký problém související s emisemi. Nicméně vedle nich existuje na trhu i poměrně malé

množství přístrojů, které umí ovládat pokročilé funkce automobilu. Nicméně tyto pokročilé přístroje jsou obvykle určeny pouze pro automobily od jednoho výrobce.

2.1.1 Zásuvka diagnostiky

Mechanické parametry zásuvky diagnostiky jsou definované normou ISO 15031-3[7]. Tvar zásuvky a čísla jednotlivých pinů jsou na obrázku 2.1. Zásuvka také musí být uživatelsky přístupná. V praxi to vypadá tak, že je zásuvka buď přístupná volně, nebo je ukrytá za nějakou formou krytu, ale vždy je přístupná bez použití náradí. Nachází se vždy v blízkosti palubní desky či sedadla řidiče.



Obrázek 2.1: Diagnostická zásuvka

Zásuvka obsahuje 16 pinů ve dvou řadách po 8 pinech. Zapojení jednotlivých pinů je v tabulce 2.1. V seznamu je překvapivé množství nespecifikovaných pinů. U nich není určena funkce a výrobce vozidla je může využít podle svého uvážení. Zároveň piny slouží pro případné rozšíření norem do budoucna, kde může dojít k jejich obsazení dalším typem komunikačního vedení. Znovu opakuji, že v konektoru nemusí být, a obvykle nejsou, osazena všechna komunikační vedení. Normy požadují, aby bylo přítomné nejméně jedno komunikační rozhraní. Které rozhraní to bude, je na výrobci vozidla.

Tabulka 2.1: Zapojení zásuvky diagnostiky.

| číslo pinu | funkce |
|------------|-----------------------------|
| 1 | nespecifikováno |
| 2 | J1850 - pozitivní vodič |
| 3 | nespecifikováno |
| 4 | kostra vozidla |
| 5 | signálová zem |
| 6 | ISO 15765-4 vodič H (CAN-H) |
| 7 | ISO 9141-2 vodič K |
| 8 | nespecifikováno |
| 9 | nespecifikováno |
| 10 | J1850 - negativní vodič |
| 11 | nespecifikováno |
| 12 | nespecifikováno |
| 13 | nespecifikováno |
| 14 | ISO 15765-4 vodič L (CAN-L) |
| 15 | ISO 9141-2 vodič L |
| 16 | Palubní napětí +12V |

2.1.2 Formát komunikace

Nyní si v krátkosti představím komunikační sběrnice používané pro diagnostiku automobilů. Popis se omezuje pouze na základní parametry jednotlivých sběrnic. Pro více informací doporučuji pročíst uvedené normy.

ISO 9141-2

ISO 9141-2[10] je sběrnice speciálně navržená pro komunikaci s emisními jednotkami vozidel s palubním napětím 12V. Obsahuje 2 vodiče označované jako K a L. Signály využívají palubního napětí. Hodnota 0 je reprezentována napětím +12V, hodnota 1 je reprezentována pomocí napětí -12V.

Komunikace probíhá sériovým způsobem. Linka K přenáší data směrem od vozidla k diagnostickému zařízení, linka L přenáší data v obráceném směru. Při komunikaci po sběrnici je třeba do hlavičky nastavit adresu jednotky, se kterou chceme komunikovat. Protokol umožňuje maximální komunikační rychlost 10,4 Kb/s.

J1850

Protokol existuje ve 2 verzích. Verze VPW využívá jenom jeden vodič. Data na komunikační lince se kódují do různé šířky impulsu. Obvykle se používá napětí 7V. Verze protokolu PWM využívá 2 vodiče a data kóduje pomocí různé střídy signálu při pevné frekvenci. Maximální rychlost komunikace je 41,6Kb/s. Návrh fyzické vrstvy protokolu umožňuje značnou odolnost proti rušení a zotavení po vážných poruchách. Například ani po zkratu na komunikačním vedení by nemělo dojít k poškození jednotek. Komunikace se automaticky obnoví jakmile bude zkrat odstraněn.

Komunikace probíhá sériovým způsobem. Při komunikaci je třeba znát adresy jednotek se kterými se komunikuje a správnou adresu nastavit do hlavičky zprávy.

ISO 15765-4

Tato komunikační sběrnice je spíše známá pod názvem CAN [9]. Sběrnice je často použita i pro propojení řídicích jednotek mezi sebou, nebo na propojení jednotek se snímači a akčními členy. Sběrnici vyvinula původně firma Bosh a stále se používá v průmyslové automatizaci. Vzhledem k úspěšnému použití této sběrnice v průmyslu, došlo k rozšíření i do automobilů. Norma definuje pro emisní diagnostiku komunikační rychlost 250 Kb/s nebo 500 Kb/s.

Při srovnání s ostatními sběrnicemi prezentovanými výše je architektura CAN sběrnice mnohem složitější. Samotná sběrnice je tvořena pouze 2 vodiči. Ovšem každá jednotka připojená na sběrnici musí mít budič sběrnice, řadič a komunikující procesor. Budič sběrnice má na starosti fyzickou vrstvu komunikace, zatímco řadič implementuje linkovou vrstvu. Sběrnice je často použita ve vozidlech koncernu Volkswagen. Protože se sběrnice používá i pro komunikaci mezi senzory, řídicími jednotkami a akčními členy, může být ve vozidle i více samostatných sběrnic.

2.1.3 Kódování chyb

Kódy chyb, které mohou být přečteny diagnostickým zařízením z emisních jednotek vozidla musí odpovídat normě ISO 15031-6 [8]. Chybový kód musí být složen z 5 znaků. První znak určuje oblast, které se kód týká:

B Výbava vozidla. Například klimatizace, airbag, světla, ...

C Karoserie. Například tlumiče, posilovač řízení, ABS, ...

P Pohon vozidla. Například motor, převodovka, emisní systémy, palivové vedení, ...

U Komunikace. Například závady na komunikačních sběrnících.

Druhý znak říká, jestli daný kód pochází z normy ISO 15031-6, nebo se jedná o kód specifický pro výrobce. 0 znamená normovaný kód, 1 znamená kód určený výrobcem.

Třetí znak určuje blíže kategorii problému pro normované kódy:

1 a 2 Měření paliva a vzduchu.

3 Zapalování.

4 Pomocné emisní systémy.

5 Kontrola a měření rychlosti.

6 Počítačové systémy.

7 a 8 Převodová skříň.

Další 2 znaky již označují konkrétní chybu.

Kapitola 3

Návrh

Úkolem práce je vytvořit prototyp zařízení schopný nahradit palubní počítač ve vozidle a provést základní diagnostiku vozidla. Provozní údaje vozidla je třeba zobrazovat řidiči za jízdy. Zařízení bude všechny potřebné údaje získávat komunikací s vozidlem pomocí diagnostické zásuvky.

3.1 Koncepce zařízení

Koncepce zařízení musí vycházet z požadavků na funkčnost. V tomto případě je důležitý požadavek na univerzální funkčnost v různých automobilech. V kapitole 2.1 jsem ukázal jaké protokoly se mohou ve vozidle vyskytovat. Tyto protokoly se od sebe výrazně odlišují. Mají různé úrovně napětí i různé typy přenosu informace. Normy pro zásuvku diagnostiky vyžadují alespoň jedno komunikační rozhraní. Bohužel nechávají na výrobci rozhodnutí, které to bude. Proto navrhované zařízení musí implementovat všechna rozhraní.

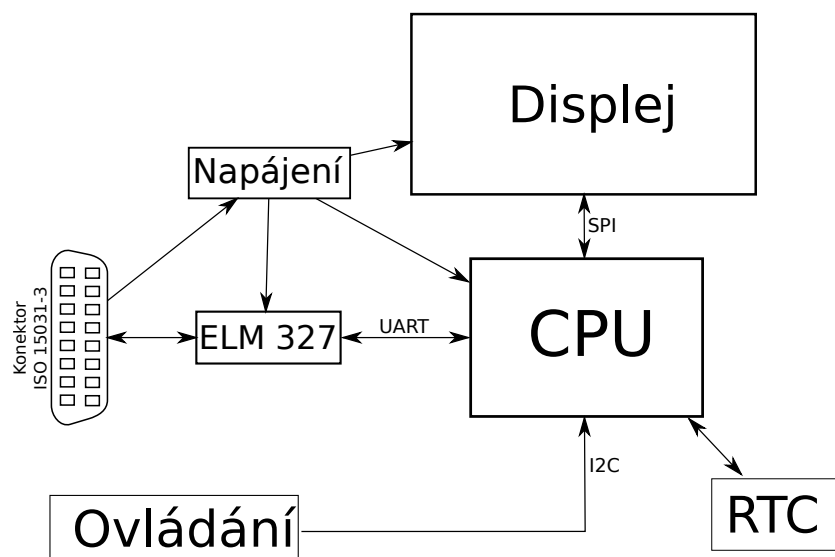
Pro takový případ naštěstí existuje řešení v podobě obvodu *ELM327* od výrobce *Elm Electronic*. Tento obvod má již implementované všechny potřebné protokoly pro komunikaci s vozidlem. Komunikace s procesorem vestavěného zařízení je pak realizována pomocí rozhraní RS232. Obvod ELM327 tedy bude základním stavebním kamenem od něhož se odvíjí další návrh zařízení. Podrobný popis obvodu najdete v kapitole 4.

Koncepce navrhovaného obvodu je nejlépe vidět na blokovém schématu na obrázku 3.1. Schéma ukazuje jednotlivé stavební bloky zařízení. Propoje mezi prvky značí komunikaci s uvedením pravděpodobného protokolu pro komunikaci. Nyní popíši jednotlivé prvky obvodu z blokového schématu.

3.1.1 Napájení

Napájení potřebuje každé elektronické zařízení. Jelikož se navrhované zařízení bude používat ve vozidle, připadají v úvahu pouze 2 řešení. Napájet celé zařízení z baterií, nebo využít palubní napětí 12V z rozvodů ve vozidle. Situace je jednodušší v tom, že palubní napětí je přítomné na pinu 16 diagnostického konektoru, viz tabulka 2.1. Pro napájení hlavní části zařízení tedy bude využito palubní napětí vozidla.

Napětí autobaterie a tím i palubní napětí vozidla je teoreticky 12V. Nicméně toto napětí se může lehce pohybovat s úrovní nabití baterie a při jízdě je typicky vyšší. V napájení se také může vyskytnout částečné zvlnění pocházející od alternátoru vozidla. Také se mohou objevit napěťové špičky indukované do vedení nebo pocházející od spínání cívek relé. Tyto rušivé signály je třeba odfiltrovat.



Obrázek 3.1: Blokové schéma obvodu.

Navíc napětí 12V je příliš velké. Navrhované zařízení bude používat napájecí napětí 5V. Pro změnu napájecího napětí z 12V na 5V se nabízí hned celá řada možných řešení. Jednoduchý napěťový dělič nepřidá v úvahu. Nebyl by schopen odfiltrovat rušení z napájecího napětí a nemohl by adekvátně reagovat na změny proudu odebíraného zařízením. Další možností je malý spínaný zdroj. Ten se vyznačuje vysokou efektivitou a galvanickým oddělením napájecích napětí. Bohužel právě galvanické oddělení napájecího napětí od vozidla je překážkou pro navrhované zařízení. Pokud by bylo napětí oddělené od vozidla, bylo by složité realizovat komunikaci zařízení s vozidlem. Nabízí se proto třetí možnost, použití regulátoru napětí.

Regulátory napětí umožňují práci pouze se stejnosměrným napájením a na rozdíl od spínaných zdrojů mohou vstupní napětí pouze snižovat. Výstupní napětí není galvanicky oddělené od vstupního. Velmi zjednodušený princip funkce regulátoru napětí je regulace vlastního vnitřního odporu. Regulátor mění svůj vnitřní odpor tak, aby úbytek napětí na regulátoru snížil vstupní napětí na požadovanou úroveň. Detailní popis obvodů pro regulaci napětí není cílem této práce.

V navrhovaném zařízení bude třeba pouze napájecí napětí 5V. Proto jsem zvolil regulátor s pevným napětím 5V, který se prodává pod označením 7805. Je to velmi oblíbený obvod, běžně užívaný pro tyto účely. Z toho důvodu je také dobře dostupný.

3.1.2 ELM327

Tento obvod tvoří základ celé konstrukce zařízení. Umí komunikovat pomocí všech protokolů používaných ve vozidlech. Pro komunikaci s procesorem v zařízení je určeno rozhraní RS232. Protože tento obvod je pro zařízení velmi důležitý, je jeho detailní popis v kapitole 4.

3.1.3 Displej

Zařízení musí získané údaje vhodně zobrazit uživateli. Lze vybírat ze 3 typů displejů. Prvním je segmentový displej, který může zobrazovat pouze čísla. To se pro navrhované zařízení nehodí. Pak zbývá rozhodnutí mezi použitím maticového displeje, který může zobrazovat

pouze text, nebo použít grafický displej.

Očekávám že zařízení bude zobrazovat větší množství údajů. Tyto údaje se navíc mohou lišit v jednotlivých vozidlech. Proto jsem zvolil raději grafický displej. Na rozdíl od textového, zde lze určit velikost písma, písmo libovolně umístit a lze vykreslit i obecnou grafiku. Grafické displeje jsou však složitější na ovládání. Obvykle je potřeba celý obraz po jednotlivých pixelech sestavit v procesoru a hotový obraz teprve odeslat do displeje. To klade větší nároky na programování centrálního procesoru a také na jeho výkon.

Pro komunikaci mezi procesorem a displejem se typicky používá rozhraní SPI. Je to velmi jednoduché sériové synchronní rozhraní. Umožňuje obousměrnou výměnu dat mezi procesorem a připojenou periferií, ale komunikaci řídí výhradně procesor. Pokud máme na rozhraní SPI připojených více periferií, je třeba zvláštním vodičem aktivovat tu periferii, se kterou se má komunikovat. Rozhraní SPI není sběrnice a nemá tedy adresy zařízení. Procesor musí sám určit se kterým zařízením bude komunikovat a to zařízení aktivovat.

Nyní už zbývá pouze rozhodnutí jestli použít displej černobílý nebo barevný. Barevné displeje dále zvyšují nároky na výkon procesoru ale oproti černobílým přinášejí lepší uživatelský zážitek. Rozhodl jsem se tedy použít černobílý grafický displej jako kompromis mezi uživatelským zážitkem, nároky na programování, nároky na výkon procesoru a cenou displeje.

Při nákupu displeje se však ukázalo, že černobílé grafické displeje napájené pomocí 5 V používají obvykle paralelní rozhraní. Oproti SPI rozhraní bude třeba větší počet vývodů z procesoru a programování komunikačního protokolu bude náročnější než v případě SPI. Nicméně oproti displejům s SPI rozhraním je výrazně levnější.

3.1.4 Procesor

Procesor bude celé zařízení řídit a provádět potřebné výpočty. K procesoru jsou připojené všechny periferní zařízení. Obvykle také vyžaduje kvalitní napájení. V napájecím napětí by neměly být žádné špičky, napětí by se nemělo vlnit. Napětí by mělo být stabilní a zdroj by měl být připraven na prudké změny v odběru procesoru.

Procesory pro podobná zařízení vyrábí celá řada firem. Také téměř všechny firmy vyrábějí procesory určené pro různé oblasti. Výsledkem je velká konkurence a tím dobrá dostupnost a cena těchto obvodů. Jednotliví výrobci si silně konkurují a v rámci této konkurence se všichni snaží vyrábět kompletní řadu od jednoduchých a levných čipů až po velmi dobře vybavené procesory s vysokým výkonem.

Navrhované zařízení na výběr procesoru neklade příliš velké nároky. Je třeba aby procesor měl na čipu rozhraní pro RS232 a I2C. To splňují takřka všechny. Výpočetní výkon také není třeba velký. Zařízení nebude provádět žádné velmi náročné výpočty, bude spíše komunikovat s okolními periferiemi. Můj odhad je, že procesor s taktovací frekvencí okolo 16MHz bude mít naprosto dostatečný výkon pro navrhované zařízení.

Vzhledem k nízkým nárokům na procesor je použitelné velké množství procesorů od nejrozličnějších výrobců. Vybral jsem si tedy firmu Atmel, protože s jejich procesory mám dobré zkušenosti a mám k dispozici vybavení pro programování jejich procesorů. Co se týká konkrétního modelu, je vhodné se podívat na výrobní řadu MegaAVR. Z této řady by byl vhodný například procesor ATmega1281. Tento procesor je dobře dostupný a splňuje všechny požadavky, které na něj klade návrh zařízení.

3.1.5 RTC

RTC značí integrovaný obvod pro uchování reálného času. Takový obvod má vlastní zdroj taktovací frekvence a sleduje čas. Pro případ výpadku napájení, nebo odpojení zařízení, má záložní baterii, která udrží obvod v chodu. Obvod se chová jako hodiny, procesor si může pomocí komunikace s obvodem RTC vyžádat aktuální čas. Navíc tento čas je udržován i po odpojení zařízení od vozidla. Pokud by byl čas sledován v procesoru, nebylo by možné udržet aktuální čas po odpojení od napájecího napětí. Jako vhodný obvod jsem vybral čip s označením DS1302.

Vybraný obvod má možnost 2 napájecích napětí. Jedno hlavní a druhé záložní. Obvod je vždy napájen tím napětím, které je větší. Jako záložní napájení lze použít baterii s napětím 3.3V. Jako hlavní napětí bude použito 5V napájení, stejné jako pro procesor a ostatní části zařízení.

Aktuální čas může být zobrazen řidiči za jízdy a může sloužit jako stopky.

3.1.6 Ovládání

Ovládání je řešeno jako samostatný modul, který bude s hlavní částí zařízení propojen co nejmenším množstvím drátů. Počítá se s umístěním ovládání na volant vozidla tak, aby mohl řidič pohodlně ovládat zařízení. Ovládání bude mít formu několika tlačítek. Stisk jednotlivých tlačítek se bude přenášet komunikační sběrnici k hlavnímu procesoru.

Protože ovládání bude propojené s hlavní částí zařízení kabelem, není třeba řešit jeho napájení. Očekává se velmi malý odběr zařízení. Proto lze vést napájecí vodiče z hlavní desky. Celý modul bude většinu času v režimu nízké spotřeby a jenom při stisku tlačítka zpracuje požadavek a odešle ho do hlavní části zařízení. Pak může opět přejít do režimu nízké spotřeby.

Očekávám, že v ovládání bude umístěn další procesor, na který budou napojena tlačítka. Na procesor nejsou kladeny nároky na výpočetní výkon. Hlavní požadavek na procesor v ovládání je jeho nízká spotřeba. K tomu účelu lze využít procesor ATmega88 od firmy Atmel. Tento procesor má několik úsporných režimů. Má však jenom 4KB paměti pro program. V případě ovládání bude program velmi jednoduchý a do 4KB se vejde s velikou rezervou.

Kapitola 4

Obvod ELM327

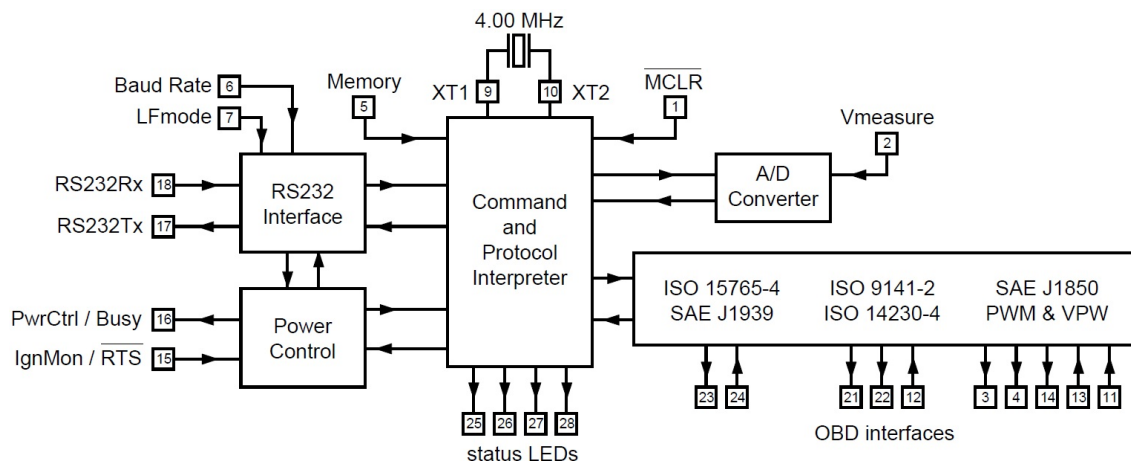
Obvod ELM327 je nejdůležitější prvek konstrukce zařízení. Proto uvedu detailnější popis jeho funkce. Všechny informace v této kapitole vychází z popisu výrobce [5].

Obvod vyrábí a dodává společnost *Elm Electronics*. Ve své podstatě se jedná o procesor, který zajišťuje překlad interního komunikačního protokolu na jednotlivé sběrnice pro komunikaci s řídicími jednotkami vozidla.

Obvod existuje v několika verzích, já se při popisu zaměřím na verzi 1.4b. Výrobce má v nabídce ještě verzi 1.3, která se od verze 1.4 liší absencí úsporných režimů. Schopnosti komunikace s vozidlem jsou stejné. Nejnovější verze 2.1 naopak vylepšuje obvod pro lepší odezvy a lepší rychlosti komunikace po RS232. Také dále vylepšuje komunikaci s CAN sběrnicí.

4.1 Princip funkce a zapojení

Obvod ELM 327 je integrovaný obvod, který umožní ovládání některé z podporovaných sběrnic pomocí rozhraní RS232. Blokové schéma obvodu je na obrázku 4.1. Z blokového schématu je dobře vidět základní princip a funkce obvodu. Zároveň je z blokového schématu vidět, že obvod umožňuje i měření palubního napětí. To je také užitečný provozní údaj.



Obrázek 4.1: Blokové schéma obvodu ELM327, převzato z [5].

Celý čip pracuje na frekvenci 4MHz a umožňuje komunikaci po sběrnicích podle norem

ISO 15765-4, ISO 9141-2 a SAE J1850. Také má výstup pro 4 stavové diody. Jako drobný doplněk je v obvodu i A/D převodník pro měření palubního napětí.

Požadavky do obvodu přicházejí po rozhraní RS232. Obvod požadavky přijme a zpracuje je v jednotce nazvané *Command and protocol interpreter*. Jednotka potom generuje signály na zvolené sběrnici. Pokud přijdou data po sběrnici z vozidla, obvod je zpracuje a předá pomocí RS232 dále do aplikace. Obvod pracuje s napájecím napětím 5V.

4.1.1 Doporučené zapojení

Výrobce doporučuje zapojení obvodu uvedené na obrázku 4.2.

Zapojení je na první pohled docela složité. Nicméně dá se rozdělit na několik částí. V pravém horním rohu je část pro zajištění napájení obvodu. U této části se předpokládá, že si ji každý přizpůsobí svým požadavkům. Důležité je, aby byl obvod napájen napětím 5V a aby zemnění obvodu bylo připojené na kostru vozidla.

Hned pod napájením je zesilovač a ochranné prvky pro rozhraní RS232. Tato část je nutná při komunikaci s počítačem, který na portu RS232 používá větší napětí než 5V. Větší napětí by poškodilo obvod. Nicméně pokud se bude komunikovat s procesorem napájeným stejným napětím, lze opět tuto část vynechat.

Po levé straně jsou pak obvody pro napojení na sběrnice v automobilu. Hned nahoře je rozhraní pro CAN sběrnici. Zde je důležité použít čipu MCP2551, který tvoří budič sběrnice a zajišťuje tak fyzickou vrstvu komunikace.

Ostatní rozhraní neskrývají velké překvapení, obvody mají za úkol pouze přizpůsobit komunikační úrovně a ochránit obvod ELM327 před poškozením napětím větším jak 5V. Tyto obvody doporučuji dodržet.

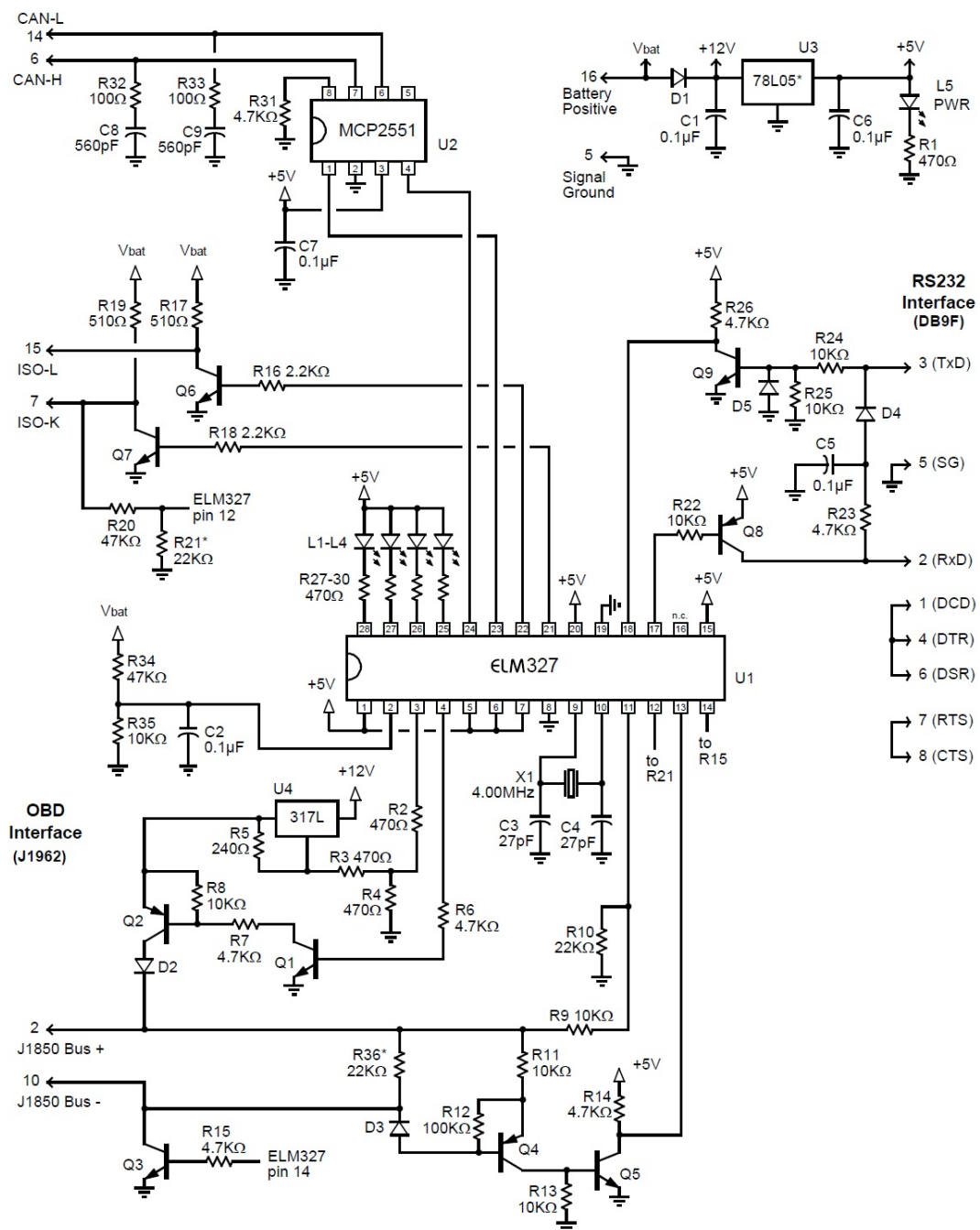
Povšimněte si také odporového děliče připojeného na pin 2. Jedná se měření napětí baterie. Napětí je děličem upraveno tak, aby nepřesahovalo 5V a dalo se bezpečně zpracovat pomocí A/D převodníku v obvodu. Reálná hodnota naměřená převodníkem je pak opět matematicky upravena tak, aby vyjadřovala napětí baterie.

4.2 Komunikace s obvodem

Obvod předpokládá komunikaci s počítačem pomocí rozhraní RS232. V případě zařízení navrhovaného v této práci bude komunikovat s procesorem po stejném rozhraní. Navíc bude celá aplikace napájena z jedné napájecí větve, což zjednoduší zapojení komunikačního vedení o převodníky napěťové úrovně.

Obvod předpokládá komunikaci na RS232 s baudovou rychlostí 9600 baudů, nebo volitelně 38400 baudů. Rychlost se vybírá pomocí nastavení logické úrovně na pinu 6. Formát komunikace je textový. Pomocí RS232 se přenáší 8 bitů užitečné informace, což odpovídá jednomu znaku podle ASCII tabulky. Veškerá komunikace je tedy textová, kódovaná do binární podoby pomocí obvyklé ASCII tabulky. V dalším textu tedy nebudu speciálně zmiňovat že se komunikuje po RS232, ale ukážu formát komunikace v textovém vyjádření.

Po zapnutí napájení obvod vypíše uvítací zprávu *ELM327 v1.4b*. Za uvítací zprávou následuje znak nového řádku a znak >. Ten značí připravenost obvodu na příjem příkazů. Příkaz se může skládat z více znaků, ale musí být vždy ukončen znakem nového řádku (0D v hexadecimálním vyjádření).



Obrázek 4.2: Doporučené zapojení obvodu ELM327, převzato z [5].

4.2.1 Interní příkazy

Interní příkazy jsou určené pro ovládání samotného obvodu ELM327, nikoliv ke komunikaci po sběrnicích. Interní příkazy vždy začínají znaky *AT*. za nimi může být nepovinná mezera a následuje název příkazu. Tabulka 4.1 obsahuje nejdůležitější interní příkazy.

Tabulka 4.1: Interní příkazy (AT).

| Příkaz | Význam |
|--------|---|
| Z | reset obvodu |
| I | vytiskne verzi obvodu |
| LP | přepne obvod do režimu snížené spotřeby |
| D | nastaví všechny volby na výchozí stav |
| BRD | nastaví baudovou rychlost pro komunikaci |
| WS | probudí obvod z úsporného režimu |
| CV | nastavuje kalibrační údaj pro přepočet napětí baterie |
| RV | vytiskne změřené napětí baterie |
| FE | vymaže všechny čekající události z paměti |
| SP | nastavení komunikačního protokolu |

4.2.2 Zaslání OBD příkazů

Pokud příkaz nezačíná znaky *AT*, pak je považován za komunikační příkaz. Obvod zkontroluje, že se jedná o platné hexadecimální znaky, příkazy zpracuje a odešle vozidlu pomocí komunikační sběrnice. Pak čeká na odpověď od vozidla a odpověď opět převede do hexadecimálního zápisu a vypíše na výstup. Stále platí, že každý příkaz musí být ukončen znakem nového řádku. Teprve po přijetí znaku nového řádku obvod začne příkaz zpracovávat.

Příkazy se tedy zasílají v binární podobě, vyjádřené jako hexadecimální znaky. mezi jednotlivými bajty je možné nepovinně zasílat mezery.

Komunikace s vozidlem může mít několik režimů. Režimy jsou uvedené v tabulce 4.2. Obvykle je třeba zvolit režim podle tabulky a pak zaslat další parametry upřesňující akci. Z toho důvodu jsou příkazy obvykle složeny ze 2 bajtů. První bajt (2 hexadecimální znaky) připadá na volbu režimu a druhý na volbu rozšiřujícího parametru.

Tabulka 4.2: Režimy příkazů [8].

| Číslo režimu | Význam |
|--------------|--|
| 01 | Aktuální data o provozu vozidla |
| 02 | Zobrazení zaznamenaných vážných chyb |
| 03 | Zobrazení méně důležitých diagnostických zpráv |
| 04 | vymazání uložených chyb |
| 05 | výsledky testů pro trvale sledované veličiny |
| 06 | provedení testů pro ostatní součásti |
| 07 | zobrazení čekajících chybových kódů |
| 08 | speciální režim |
| 09 | informace o vozidlu |
| 0A | vytiskne permanentní chybové kódy |

Odpovědi od vozidla obvod vytiskne ve formě hexadecimálních znaků, kde je každý bajt oddělen mezerou. V odpovědi vždy první bajt znamená potvrzení režimu na který se odpovídá. Číslo v potvrzení je číslo režimu navýšené o 40 hexadecimálně. Druhý bajt je zopakování rozšiřujícího parametru bez úpravy. Pak již následují data odpovědi. Jejich délka se může lišit. Po ukončení odpovědi obvod vloží znak konce řádku.

Odpovědi na jednotlivé příkazy mají dané formáty, ale příkazů je mnoho a tak popis všech možných formátů odpovědi je mimo rozsah této práce.

Pomocí interních příkazů také lze zapnout hlavičky zpráv. Běžně obvod vypisuje pouze datové oblasti zprávy ze sběrnice, ale lze zapnout výstup všech dat dané zprávy. To je užitečné zvláště když chceme zjistit která jednotka na daný příkaz odpovídá, protože v hlavičce zprávy bude její adresa na sběrnici.

4.2.3 Pokročilá komunikace

Obvod ELM327 umožňuje také pokročilejší komunikaci. V tomto případě si můžeme zvolit přímo komunikační protokol a pak zasílat binární data (v hexadecimálním zápisu), která obvod následně vyšle na zvolenou sběrnici. Tímto způsobem je možné realizovat libovolnou komunikaci po sběrnici, ovšem je třeba znát přesně formát komunikace na dané sběrnici. Takové nenormované příkazy jsou také obvykle specifické pro daného výrobce a model vozidla.

Obvod umožňuje také monitorování vybrané sběrnice, kdy zasílá na výstup veškerá data, která na sběrnici byla. Nehledí se přitom na příjemce z hlaviček zprávy na sběrnici. Dá se tak sledovat veškerá komunikace na sběrnici vozidla, i taková, která nebyla určena obvodu ELM327. Zvláště v případě CAN sběrnice to může být zajímavý nástroj na pochopení funkce řídicích jednotek. Samozřejmě za předpokladu že řídicí jednotky používají ke komunikaci se senzory sběrnici, která je vyvedena do diagnostického konektoru. Sběrnic může mít vozidlo i více a ne všechny musí být vyvedené do diagnostického konektoru.

Pokročilá komunikace se pravděpodobně nebude v navrhovaném zařízení používat a tak její popis už je mimo rozsah této práce.

Kapitola 5

Návrh schématu

Během návrhu schématu je vhodné dodržovat rozdělení jednotlivých funkčních částí z blokového schématu. Tím se dají vytvořit skupiny součástek, které spolu opravdu funkčně souvisí. Pozornost čtenáře pak není rozbita mezi vzájemně nesouvisející součástky. Schéma je tím přehlednější, zejména pro nezúčastněného čtenáře. Proto je schéma rozděleno na 3 samostatné listy a dále členěno do funkčních bloků, které jsou mezi sebou propojeny pomocí odkazů. Názvy odkazů pak pojmenovávají jednotlivé signály. V této kapitole budou postupně rozebrány všechny části návrhu schématu a jejich účel ve výsledném zařízení. Popis se soustředí zejména na účel jednotlivých součástek pro funkci nebo vývoj zařízení. Nejsou však uváděny hodnoty součástek. Výpočty hodnot součástek v tomto návrhu jsou pouze jednoduché úlohy na jejichž zopakování stačí základní znalosti slaboproudé elektrotechniky. Následující text v této kapitole předpokládá, že čtenář alespoň tyto základní znalosti má a proto výpočty hodnot neuvádí. Všechny hodnoty součástek lze samozřejmě najít ve schématu obvodu.

Pro návrh schématu je použitý program KiCad. Jedná se o skupinu programů šířených zdarma pod licencí GNU GPL v2. Tento nástroj obsahuje kompletní a vzájemně provázané nástroje pro návrh schématu, návrh tištěného spoje i tvorbu dat pro výrobu výsledné elektroniky.

5.1 Připojení vozidla

Na listu schématu nazvaném "Rozhraní vozidla" je umístěno vše, co souvisí s komunikací s vozidlem a připojením. Jádrem komunikace s vozidlem je obvod ELM327, který ke své činnosti potřebuje další napěťové přizpůsobení pro jednotlivé komunikační sběrnice. Zapojení obvodu ELM327 a jednotlivých komunikačních rozhraní vychází z doporučení výrobce [5].

Konektor pro připojení je na straně vozidla dán normou. Ovšem mít na zařízení pevně připojený kabel je nepraktické. Na zařízení tedy bude také konektor pro připojení kabelu. Od vozidla je třeba přivést celkem 8 vodičů. Navíc jednotlivá vedení obvykle tvoří pár, proto je vhodné připojení provést pomocí kabelu, který obsahuje kroucené páry vodičů. Z toho důvodu jsem zvolil kabel typu category 5e a jako konektor běžnou zásuvku RJ45, ve schématu označenou jako CON4. Uvedený kabel a zásuvka se sice obvykle používají pro vedení Ethernetu, nicméně pro účely tohoto zařízení se hodí také. Navíc zásuvka RJ45 obsahuje pojistku proti náhodnému vytáhnutí kabelu, což se v prostředí vibrací za provozu vozidla hodí také.

Protože napájecí napětí je odebíráno také z vozidla, je na tomto listu umístěný i obvod

pro úpravu napájecího napětí. A právě popisem napájení zařízení začnu.

5.1.1 Napájení

Jak již bylo řečeno, zdrojem napájecího napětí je samotné vozidlo, přesněji řečeno jeho baterie respektive alternátor. Bohužel tento zdroj není dokonalý. Palubní napětí se může pohybovat v širokém rozsahu. Běžné napětí autobaterie je 12 V. Nicméně při startování, kdy je baterie silně zatížena startérem, může napětí krátce poklesnout i na 8 V. Naopak při běžícím motoru, kdy je v činnosti alternátor, je palubní napětí asi 13,5 V. V palubním napětí se také může objevit rušení v podobě indukovaných špiček pocházejících třeba od zapalování. Tyto vlivy je třeba vhodným návrhem napájecího obvodu potlačit.

Komunikační rozhraní ve vozidlech používají různé napěťové úrovně pro komunikaci. Nicméně napájení vlastních komunikačních rozhraní bude řešeno během návrhu jejich obvodů. Mimo komunikačních sběrnic je použito pouze napájení 5 V. To vyhovuje všem použitým prvkům. Zejména kvůli procesoru je třeba, aby napětí bylo stálé, bez výrazných výkyvů. Celé zařízení je koncipováno jako digitální mikroprocesorem řízené zařízení. To znamená, že odběr zařízení bude mít spíše impulsní charakter. Impulsy vznikají spínáním v digitálních obvodech. To klade zvýšené nároky na tvrdost zdroje.

Z bezpečnostních důvodů, hned první součástka za vstupem napájení do zařízení musí být pojistka. Ve schématu označena F1. Byla vybrána standardní tavná trubičková pojistka, 0,5 A. Další součástkou už je napěťový regulátor.

Po úvaze jsem vybral pro napájení napěťový regulátor s označením 7805. Ve schématu označen jako U7. Jedná se o napěťový regulátor s pevným výstupním napětím. Na vstupu dle výrobce může být široký rozsah napětí. Obvod může spolehlivě pracovat až do 35 V na vstupu [6], což je pro účely napájení z palubní sítě automobilu více než dostatečné. Zároveň velmi rychle reaguje na změny napětí na vstupu a změny odběru na výstupu tak, aby na výstupu bylo dodrženo stanovené napětí 5 V. Obvod ovšem nezajišťuje galvanické oddělení a přebytečné napětí se mění na teplo. Při větším odběru se může regulátor dosti zahřívat.

Regulátor ke správné funkci potřebuje ještě kondenzátory na vstupu a výstupu. Ty slouží zejména ke stabilizaci regulátoru, který je bez nich náchylný na kmitání [6]. Tyto kondenzátory jsou označeny C13 a C14. Kondenzátory také pomáhají regulátoru lépe zvládat impulsní charakter spotřeby.

5.1.2 Sběrnice CAN

Pro komunikaci po sběrnici CAN se používá fyzická vrstva implementovaná v obvodu MCP2551, ve schématu jako U4. Použití tohoto obvodu značně zjednodušuje další návrh. Na komunikačních vodičích před tímto obvodem je ukončení linky pomocí RC členu, ve schématu jako rezistory R7, R10 a kondenzátory C9 a C10. Pak již stačí jenom podpůrné součástky jako je R4 a C8. Obvod se napájí z napájecí větve 5 V.

5.1.3 Sběrnice ISO 9141-2 a ISO 14230

Tyto 2 sběrnice jsou si velmi podobné. Hlavně používají stejnou signálovou úroveň 12 V. Jako napájecí větev je tedy použitý přímo napájecí přívod od baterie bez dalších úprav. Je třeba si uvědomit, že komunikaci ovládá obvod ELM327, který je ovšem napájen napětím 5 V. Je tedy třeba přizpůsobit napěťové úrovně pro vysílání i příjem a tím ochránit ELM327 před poškozením vyšším napětím.

Komunikace po sběrnici se provádí způsobem zvaným jako otevřený kolektor. To znamená, že klidová úroveň sběrnice je v logické 1. Klidovou úroveň udržují pull-up rezistory R23 a R24. Pro vyslání logické 0 po sběrnici, je třeba vodiče uzemnit pomocí tranzistorů Q6 nebo Q7. Rezistory R25 a R26 jsou zde pouze pro omezení proudu bází tranzistoru. Tento způsob je velmi výhodný právě vzhledem k přizpůsobení napěťových úrovní. Tranzistor může spínat napětí větší než napájecí napětí řídicího obvodu, ale na jeho sepnutí stačí i menší napětí.

Pro čtení dat je třeba přizpůsobit napěťovou úroveň z 12 V zpět na 5 V. K tomu slouží napěťový dělič z rezistorů R21 a R22. Napěťový dělič je nastaven tak, aby na jeho výstupu byla přibližně třetina napětí vstupu. To umožní bezpečný provoz do 15 V na vstupu. Pro komunikaci v automobilu je taková úroveň bezpečná.

5.1.4 Sběrnice J1850

Tato sběrnice má ze všech navrhovaných nejsložitější fyzickou vrstvu. Existují 2 varianty komunikace. Režim VPW, kde se komunikuje po jednom vodiči se signálovou úrovní 7 V. Druhý je režim PWM, kde se komunikuje po diferenciálním páru vodičů se signálovou úrovní 5 V.

V rámci návrhu schématu pro tuto sběrnici je tedy třeba zajistit zdroj napětí, který bude možné přepínat mezi napěťovou úrovní 5 V a 7 V. To zajišťuje regulátor U5 s rezistory R14 a R16 až R18. Pomocí zvláštního vstupu pak probíhá přepínání mezi uvedenými úrovněmi.

Řízení kladného pólu sběrnice pak probíhá pomocí dvojice tranzistorů Q2 a Q5 a souvisejících rezistorů R9, R12 a R20. Dioda D3 pak pouze chrání zdroj a spínací obvod pro případ že by se na sběrnici bylo větší napětí, než je nastavené na zdroji. Řízení záporného pólu sběrnice zajišťuje pouze tranzistor Q1 a R3 omezující proud bází tranzistoru.

Snímání stavu vodičů na sběrnici je pak rozdělené podle toho, jaký režim se použije. Pro režim VPW stačí napěťový dělič realizovaný rezistory R13 a R19. Pro režim PWM pak jsou potřebné ostatní, zatím nepopsané součástky. Snímání tohoto režimu je komplikované zejména komunikací po diferenciálním páru, kde se signál na páru vodičů musí zpracovat na obdélníkový signál vstupující do čipu ELM327. Rozdíl napětí na diferenciálním páru sepne tranzistor Q3, jeho výstup je pak zesílen tranzistorem Q4, který tak vytvoří obdélníkový signál s rychlou změnou stavu. Rezistory slouží jen pro omezení proudu přes tranzistory a ochranná dioda D2 chrání obvod proti přepólování sběrnice.

5.1.5 Obvod ELM327

K obvodu jsou samozřejmě připojeny převodníky úrovně popsané výše. Připojení a parametry převodníků jsou podle doporučení výrobce. K tomu je třeba připojit napájení 5 V. Jako u všech digitálních integrovaných obvodů, je velmi vhodné připojit mezi napájecí piny filtrační kondenzátor, zde C15. Filtrační kondenzátor má pomáhat potlačovat rušení na napájecím vedení a potlačovat rušení způsobené impulsním charakterem odběru součástky.

Dále jsou použité signalizační diody D4 až D7 společně s omezovacími rezistory R27 až R30. Diody složí k signalizaci činnosti uživateli. 2 diody jsou pro signalizaci vysílání a příjmu pro komunikaci s procesorem. Další 2 diody slouží pro signalizaci komunikace s vozidlem. Diody plní pouze signalizační funkci, jejich vynechání nemá vliv na funkci obvodu.

Důležitou součástí pro funkci obvodu je však taktovací krystal X2. Taktovací krystal pro obvod ELM327 má být naladěný na frekvenci 4 MHz. Krystal je zatížen kondenzátory C11 a C12 podle doporučení výrobce krystalu.

Od obvodu jsou pak vedeny komunikační signály k procesoru, společně se signály pro nastavení chování obvodu a resetovací signál. Tyto signály by bylo také možné připojit přímo na napájecí napětí nebo na zem. Nicméně toto řešení dovoluje ovládat vstupy pomocí programu z hlavního procesoru a zapojení je tak univerzálnější. Zejména vedení signálu reset k procesoru je důležité, protože to umožňuje resetovat obvod ELM327 až ve chvíli, kdy na to je řídicí program v procesoru připraven.

5.2 Zapojení periférií

Na listu schématu pojmenovaném jako "Periferie" je zapojení pouze 2 periferních obvodů k procesoru. Je to obvod pro udržování reálného času, dále v textu označovaný jako RTC obvod. Druhým, důležitějším, obvodem je displej.

5.2.1 RTC obvod

Jako obvod reálného času je použitý obvod DS1302, ve schématu označený jako U8. Pro správnou funkci obvodu je třeba jenom připojit správný taktovací krystal a napájení. Taktovací krystal musí mít frekvenci 32,768 kHz. Zatěžovací kondenzátory krystalu jsou již integrované v obvodu, takže v tomto případě se žádné další nepřipojují. Při výběru krystalu je třeba vybrat typ laděný na zatěžovací kapacitu 6 pF. Přesně takový krystal doporučuje výrobce čipu [4].

Dále je třeba připojit napájení. Obvod reálného času má zvláštnost v podobě 2 napájecích vstupů. Jeden napájecí vstup je určen pro běžné napájení z větve 5 V, která je použita i pro napájení procesoru. Na napájecí vstup je opět vhodné připojit filtrační kondenzátor, ve schématu označený C17. Druhý napájecí vstup je určen pro zálohovací baterii. Napájení obvodu se automaticky přepne na baterii v případě, že dojde ke ztrátě hlavního napájení. Bateriové napájení slouží jen k udržení aktuálního nastavení a k udržování běžícího času. Dokud je obvod napájený z baterie, nelze s ním komunikovat [4]. To v případě navrhovaného zařízení nevádí, neboť hlavní procesor je napájen ze stejné napěťové větve. V případě výpadku napájení tak nepoběží ani hlavní procesor a tedy nebude potřebovat komunikovat s obvodem RTC.

Pak už zbývají pouze piny sloužící ke komunikaci. Vstupní piny jsou připojené přímo k procesoru, ale pin pro data je připojen pomocí rezistoru R35 v sérii. Rezistor slouží pouze jako bezpečnostní opatření pro případ chybné implementace komunikačního protokolu. Pokud by pak datové vedení bylo aktivně buzeno ze strany procesoru i RTC obvodu, rezistor v sérii zajistí obvody před poškozením vysokým proudem.

5.2.2 Displej

Displej je v zařízení jediný prostředek, pro zobrazení zjištěných údajů uživateli. Je tedy pro funkci celého zařízení nepostradatelný a při návrhu jeho zapojení je třeba dbát zvýšené opatrnosti. Zároveň se jedná o nejdražší součástku v celém zapojení.

Z displeje vede celkem 20 vodičů. Hned krajní 2 vodiče slouží k napájení logických obvodů elektroniky displeje. Na ně se tedy připojí standardní napájení +5 V. Vstup označený V0 je napájení pro řízení jednotlivých bodů na LCD displeji. Toto napětí by mělo být při pokojové teplotě displeje okolo 8,2 V. Naštěstí displej má svůj spínaný zdroj na vývodu Vee, ze kterého je možné pomocí trimru RV1 vytvořit požadované napětí. Trimr RV1 tak slouží k nastavení kontrastu bodů na displeji [11].

Následují vývody používané pro komunikaci. Mimo vlastních datových vodičů zde najdeme i signály pro určení čísla čipu se kterým se komunikuje nebo vzorkovací signál. Důležitý je signál označený jako RSTB. Pokud je tento signál v logické 0, je displej vypnut a uveden do výchozího stavu [11].

Poslední 2 vývody slouží pro napájení diody podsvícení. Podle výrobce je již před diodou osazen vhodný omezovací rezistor a má se používat přímo napětí +5 V pro napájení [11]. Aby bylo možné ovládat podsvětlovací diodu pomocí programu z hlavního procesoru, je dioda spínána tranzistorem Q8. Přidružený rezistor R33 pak slouží pro omezení proudu bázi tranzistoru. Rezistor R34 pak zajistí spolehlivé uzavření tranzistoru a tím zhasnutí podsvícení v případě, kdy procesor nebude aktivně ovládat bázi tranzistoru.

Rezistory R36 a R37 mají za úkol udržovat displej v neaktivním stavu v případě, kdy jsou výstupy hlavního procesoru uvedené do stavu vysoké impedance. Tento stav nastává například během nahrávání nové verze programu do procesoru. Protože nahrávání nové verze sdílí některé piny procesoru s ovládáním displeje, je důležité, aby se procesor během probíhajícího přenosu programu nepokusil o žádnou komunikaci.

5.3 Zapojení procesoru

Srdcem zařízení je procesor Atmel ATmega 1281. Ve schématu je označený jako U1. Do procesoru se sbíhají takřka všechny komunikační vodiče v zařízení. Ačkoliv se jednotlivé piny dají ovládat samostatně z programu v procesoru, ne vždy je samostatné ovládání jednotlivých vodičů vhodné. Procesor obsahuje množství periferních obvodů přímo na čipu procesoru. Vývody těchto periferních obvodů jsou pak na některých pinech procesoru a mohou nahradit původní univerzální funkce [2]. Jako příklad lze uvést modul pro sériovou komunikaci po lince RS232. Takové moduly má procesor rovnou 2. Pomocí těchto modulů je možné zejména ovládání komunikačních vedení zjednodušit v programu na práci s periferní jednotkou procesoru. Využití periferních obvodů výrazně zjednodušuje programování a také šetří čas procesoru, který je možné věnovat důležitějším úkolům. Proto je důležité připojovat jednotlivé signály k procesoru na správné piny tak, aby mohlo být maximálně využíváno vestavěných periférií.

5.3.1 Napájení a zdroj hodinového signálu

Procesor ATmega 1281 má více vstupů napájení. Nejprve se tedy připojí všechny 3 vývody GND na zemnění. V navrhované aplikaci není třeba při připojování zemnění provádět nějaké speciální opatření. Jiná situace by nastala u obvodů pracujících s analogovým signálem. V navrhovaném zařízení se ale s analogovým signálem nepracuje.

Dále je třeba připojit napájení na společný napájecí rozvod 5 V. To se provede pomocí pinů VCC. U procesoru je důležité nezapomenout na filtrační kondenzátory u každého ze vstupů napájení. Tuto roli v zapojení plní kondenzátory C6 a C7. Pin AVCC slouží pro napájení vestavěných analogových obvodů. V navrhovaném zařízení se nepočítá s použitím komparátoru ani AD převodníku, takže jeho vstup se prostě připojí na napájení bez dalších zásahů. Pin AREF slouží jako vstup referenční hodnoty pro napěťový komparátor, AD převodník a brown-out detektor. Pravděpodobně bude využitý vestavěný brown-out detektor. V takovém případě se na pin AREF má připojit kondenzátor proti rozvodu země. Kondenzátor má sloužit ke stabilizaci a odrušení vnitřní napěťové reference procesoru. Tuto úlohu plní ve schématu kondenzátor C3.

Napájený procesor potřebuje také zdroj taktovací frekvence. Možností zdrojů frekvence ve vybraném procesoru je více. Taktovací frekvence přímo ovlivňuje výpočetní výkon procesoru. Proto jsem se rozhodl pro jistotu použít největší dovolenou taktovací frekvenci 16 MHz. Té se dá dosáhnout pouze použitím taktovacího krystalu [2]. Taktovací krystal pro procesor je označený X1. K němu náleží zatěžovací kondenzátory C1 a C2. Kapacitu kondenzátorů předepisuje výrobce vybraného krystalu na 18 pF. Tím je zajištěna možnost provozovat procesor na nejvyšší dovolené frekvenci. V případě, že by se během vývoje ukázalo, že není takový výkon potřeba, je vždy možnost frekvenci snížit děličkou kmitočtu vestavěnou v procesoru.

5.3.2 Ovládací vstupy

Pro ovládání zařízení budou k dispozici 4 tlačítka a ovladač určený k umístění na volant. Ovladač na volant bude mít také 4 tlačítka se stejnou funkcí jako tlačítka na zařízení. Tlačítka, ve schématu označená jako SW1 až SW3 a SW5 jsou připojená na samostatné vstupy procesoru. Na vstupy není kladen žádný zvláštní požadavek.

Spojení s modulem ovladače bude zajišťovat komunikační linka s protokolem komunikace označovaným jako I2C. Konektor pro připojení vedení od ovladače je označen jako CON1. Má 4 piny. 2 z nich slouží jako napájení ovladače napětím 5 V. Zbylé 2 vodiče jsou využité pro vedení komunikační linky I2C. V procesoru je již vestavěný modul pro komunikaci po takovém typu vedení a je vyvedený na pinech 25 a 26. Linka z konektoru je tedy připojena na tyto piny.

5.3.3 Rozhraní vozidla a periferií

Obvod hodin reálného času je připojen pomocí 3 vodičů k procesoru bez zvláštních požadavků na výbavu daných pinů. Komunikační protokol neodpovídá žádnému běžnému protokolu, který by již byl implementován formou vestavěného modulu v procesoru. Celou komunikaci včetně časování tak bude třeba provádět pomocí programu. Proto je obvod připojen na obecné vstupně-výstupní piny.

Komunikace s displejem také neodpovídá žádnému protokolu, který by již byl v procesoru implementován. Proto připojení komunikačních vodičů také neklade zvláštní požadavky na výbavu jednotlivých pinů. Ovšem vhodným připojením i tak lze zjednodušit programování komunikace. Od displeje přichází celkem 16 vodičů. Přičemž 8 z nich je paralelní datová sběrnice, ostatní jsou řídicí signály. Je proto vhodné celou datovou sběrnici napojit na jeden celý port procesoru. Tím lze ovládat všech 8 vodičů datové sběrnice pomocí jediné instrukce zápisu do jediného registru. Jinak by bylo nutné nastavovat každý bit v programu zvlášť. Proto pro připojení datových vodičů byl vybrán port E. Řídicí signály jsou pro jednoduchost také připojeny v rámci jednoho portu B.

Z připojení displeje je nejzajímavější řízení spínání LED diody. Podsvětlovací diodu by bylo možné rozsvítit a zhasnout libovolným pinem, ale v případě podsvětlení je vhodné také regulovat intenzitu svitu. Toho lze dosáhnout pomocí pulsně šířkové modulace, dále označované jako PWM. Vybraný procesor může generovat PWM signál pomocí obvodu výstupního komparátoru čítače. Nezávislých kanálů na generování PWM signálu je v procesoru více, já jsem pro připojení diody vybral pin 7 portu B. Na něm je výstup komparátoru A čítače č. 0 a zároveň komparátoru C čítače č. 1. Je tak na výběr dokonce ze 2 interních čítačů a během programování si budu moci zvolit, kterým čítačem bude řízen svit diody podsvícení.

Připojení obvodu ELM327 pro komunikaci s vozidlem je realizováno pomocí 5 vodičů. 2 vodiče jsou pro komunikaci s čipem, ostatní slouží pro jeho ovládání. Na připojení ovládacích

vodičů nejsou kladeny zvláštní nároky, lze vybrat kterýkoliv pin procesoru. Komunikační vodiče používají sériovou komunikaci pomocí protokolu RS232. Pro ten jsou v procesoru 2 různé komunikační moduly, které tento protokol podporují. Pro připojení jsem vybral modul č. 1, protože vývody modulu č. 0 jsou již obsazeny připojením displeje a navíc piny slouží i pro servisní konektor. Je tedy vhodnější vybrat piny, které nejsou obsazené jinou funkcí.

K procesoru je také připojena svítivá dioda D1 i s omezovacím rezistorem. Dioda bude sloužit pro diagnostiku zařízení a hlášení výskytu chyb v programu. Pro pin procesoru neklade žádné zvláštní nároky, ani se nepředpokládá nutnost řízení intenzity svitu.

5.3.4 Externí paměť

Asi nejvýraznější podpůrný obvod je externí paměť realizovaná pomocí čipů U2 a U3. Obvod U2 je samotná paměť, obvod U3 je 8 registrů typu D. Pomocí registru je rozšířen počet výstupů, protože rozhraní sdílí spodních 8 bitů adresové sběrnice s datovou sběrnicí. Nyní, v době návrhu, není zcela jasné kolik operační paměti bude třeba pro běh programu v procesoru. Procesor sám poskytuje 6 kB operační paměti, ale je možné paměť dále rozšířit pomocí externě připojené paměti typu SRAM [2]. Proto jsem se rozhodl při návrhu schématu připravit připojení externí paměti tak, aby na desce plošného spoje bylo vše nachystáno pro osazení paměti. Jsem ovšem připraven rozšířenou paměť osadit až v případě, že bude opravdu potřeba. V opačném případě zůstane místo pro paměť na desce plošného spoje neosazené.

Externí paměť zabírá poměrně hodně pinů procesoru, i přes částečné sdílení adresové sběrnice s datovou sběrnicí. Rozhraní obsazuje celé porty A a C a k tomu 3 piny z portu G. Kondenzátory C4 a C5 pak slouží jako filtrační kondenzátory pro napájení čipů U2 a U3. Kondenzátory budou obsazeny i v případě, že externí paměť osazena nebude. Jelikož kondenzátory dále zvyšují kapacitu napájecího rozvodu, tak z funkčního hlediska ničemu nepřekážejí a spíše pomáhají lépe stabilizovat regulátor napětí.

5.3.5 Servisní obvody

Servisní obvody mají, jak název napovídá, sloužit k servisu a vývoji zařízení. Všechny popisované obvody jsou v dokončeném zařízení již zbytečné a nepoužívají se. Nicméně pro vývoj zařízení jsou důležité.

Nejdůležitějším servisním konektorem je ISP konektor, označený ve schématu jako CON2. Procesor bude v zařízení připájen přímo na desku plošného spoje. Nebude tedy možné jeho vyjmutí ze zařízení a nahrání nové verze programu. Právě pro nahrání firmware do zařízení je osazen ISP konektor. Konektor obsahuje napájení procesoru, komunikační linku pro přenos dat a signál externího resetu procesoru. Veškeré operace při programování procesoru totiž probíhají během aktivního signálu resetu procesoru, jedině v tomto stavu je programovací rozhraní aktivní [2].

Připojení komunikační linky je pak provedeno podle doporučení výrobce procesoru. Zajímavostí je, že komunikační linka pro programování sdílí 2 piny procesoru běžně používané pro komunikaci s displejem. Proto jsou na důležitých komunikačních linkách k displeji osazeny pull-down rezistory R36 a R37. Ty mají za úkol po dobu programování procesoru udržet displej v neaktivním stavu. Pokud by se displej během programování procesoru pokusil o komunikaci, došlo by k porušení nahrávaného programu. Takto je zajištěno, že displej během programování přepne své komunikační vstupy do stavu vysoké impedance a nebude reagovat na žádná data, která se na vodičích budou vyskytovat.

Druhý servisní konektor, označený jako CON3, je připojení rozhraní JTAG k procesoru. Toto rozhraní může plnit hned několik servisních funkcí. Pomocí JTAG lze ovládat výstupy procesoru a ověřit tak, že je procesor správně připájený na desku plošného spoje a samotná deska je správně vyrobená. Dále rozhraní může sloužit pro připojení plnohodnotného debuggeru pro krokování programu. To může být při ladění chyb v programu velká výhoda. Také je možné pomocí rozhraní JTAG nahrát novou verzi firmware. Díky těmto možnostem může konektor plně nahradit konektor ISP, ale pro využití JTAG rozhraní je třeba lepší vybavení laboratoře než je tomu u konektu ISP. Navíc piny obsazené rozhraním JTAG jsou obsazené i během normální činnosti zařízení a nelze je tak sdílet pro jiný účel.

Jak v konektoru ISP, tak i v konektoru JTAG je jeden vodič vyhrazen pro signál externího resetu procesoru. Tento signál je připojen k procesoru na pin 20, který je k tomuto účelu vyhrazen. Pin pro reset sice má interní pull-up rezistor, ale tento vestavěný rezistor je poměrně slabý. Vzhledem k délce vedení, které může být připojené k pinu je vhodné dodat ještě externí rezistor, který bude bezpečněji držet klidovou úroveň pinu na úrovni napájecího napětí. K tomu slouží rezistor R1. Tlačítko SW4, které je ve schématu hned vedle pak slouží k ručnímu resetu procesoru. Ve finálním zařízení taková funkce není nikdy třeba, ale během vývoje je užitečná. Jinak by se musel reset procesoru provádět pomocí odpojení napájecího napětí.

5.4 Ovládací modul na volant

Jak již bylo zmíněno v kapitole 5.3.2, pro připojení ovládacího modulu je použitý konektor se 4 piny. 2 piny slouží pro přívod napájení a další 2 piny pro komunikaci pomocí protokolu I2C.

Základem ovládacího modulu je procesor Atmel ATmega88. Jedná se o jednoduchý procesor, téměř základní model v modelové řadě ATmega [3]. Tento základní model byl zvolen právě pro svoji cenu. Pro obsluhu ovládacího modulu není třeba mnoho funkcí ani velký výpočetní výkon. I základní model procesoru tak silně převyšuje potřeby modulu.

Kromě již zmíněného konektoru pro spojení s hlavním procesorem zařízení je osazen ještě servisní ISP konektor pro nahrání firmware. Následují tlačítka SW1 až SW4. Právě události stisku tlačítek se budou zasílat hlavnímu procesoru. Stav tlačítek se bude snímat pomocí programu, takže piny pro připojení tlačítek nemusí mít žádnou speciální výbavu.

Mimo tlačítek jsou ještě osazeny svítivé diody D1 až D3 i se svými omezovacími rezistory. Účel diod zatím není určen, nicméně by bylo možné diody použít pro hlášení událostí uživateli.

Pak už je k procesoru jenom připojeno napájení a opět filtrační kondenzátor C1. Schéma modulu pro volant tak pěkně ukazuje, že na základní použití procesoru nejsou třeba téměř žádné vnější součástky. V tomto případě není třeba připojovat ani taktovací krystal, který se jinak běžně připojuje. Modul si vystačí s interním oscilátorem na frekvenci 8 MHz. To je možné v případech kdy není důležitá stabilita kmitočtu procesoru vzhledem k okolní teplotě a stačí kmitočet 8 MHz. Procesor podporuje i větší kmitočet, tento konkrétní model dokonce až po 20 MHz, ale takový výpočetní výkon v ovládacím modulu není třeba.

Kapitola 6

Návrh tištěného spoje

Návrh tištěného spoje je proveden v programu KiCad a to s pomocí schématu vytvořeného a popsaného v předchozí kapitole. Nástroj pak umožňuje automatickou kontrolu shody navrženého plošného spoje se schématem. Tím je minimalizováno riziko vzniku chyb na úrovni návrhu tištěného spoje. Samotný postup při návrhu tištěného spoje vždy vychází ze zkušeností návrháře a funkce konkrétního obvodu. Návrh se vždy snaží najít vhodný kompromis mezi požadavky na velikost výsledné desky, snadností osazení a zapájení součástek, výrobitelnost desky a výrobní cenou desky. Tyto požadavky jdou vzájemně proti sobě.

Proto se tato kapitola zaměří hlavně zdůvodnění důležitých rozhodnutí během návrhu plošného spoje. Zejména je zmíněno zmiňuje omezení na umístění jednotlivých součástek a vedení důležitých signálů.

6.1 Příprava návrhu plošného spoje

Než se začne s návrhem tištěného spoje, je třeba určit celou řadu parametrů. Rozhodování také komplikuje fakt, že dodatečná změna některého rozhodnutí učiněného v přípravě návrhu, obvykle znamená provedení celého návrhu tištěného spoje znovu. Je tedy třeba velmi pečlivě volit jednotlivé parametry.

6.1.1 Výběr pouzder součástek

Schéma určuje typ a hodnotu součástky, často se však stává, že výrobce součástky dodává stejný obvod ve více provedeních pouzdra. Nejčastěji lze nalézt součástku ve variantě s vývody, nebo ve variantě pro plošnou montáž. Plošná montáž má často i několik variant pouzder pro stejný obvod, lišících se velikostí pouzdra i provedením přívodů.

Technologie vývodových součástek je jednodušší pro ruční osazování a pájení, ale zase vede na složitější návrh vedení spojů a obecně větší plochu desky. K velikosti plochy desky je pak přímo úměrná výrobní cena jednoho kusu.

V navrhovaném zařízení jsou použity pouzdra součástek pro povrchovou montáž. To umožní vytvořit relativně malou desku, ale ruční osazování a pájení bude složitější. U navrhovaného zařízení se počítá zatím s výrobou jediného prototypu, kde čas osazování desky není tak kritický jako u sériové výroby. Naopak velikost desky může příznivě ovlivnit výrobní cenu prototypu. Použití součástek pro plošnou montáž také zjednoduší návrh desky spojů.

U součástek jako jsou rezistory a kondenzátory volím spíše větší velikosti pouzder, obvykle s označením velikosti 0805. Tato velikost pouzder je vhodným kompromisem mezi

plochou zabranou součástkou a snadností osazení a ručního zapájení. Menší velikosti pouzder by bylo obtížné, nebo i nemožné osadit a zapájet ručně.

Vývodová technologie je použita pouze u konektorů, držáku baterie a držáku pojistky. Ostatní součástky jsou v pouzdrech pro povrchovou montáž. Výběr konkrétních pouzder součástek je v projektu pro návrhový systém KiCad. Projekt je umístěn na přiloženém CD k této práci.

6.1.2 Výběr počtu vrstev spoje

Jedno z nejdůležitějších rozhodnutí, které ovlivňuje celý návrh desky plošných spojů je počet vrstev spojů. Obecně platí, že čím větší je počet vrstev spojů v desce, tím je složitější a dražší výroba. Nicméně na větším počtu vrstev je návrh jednodušší, protože nedochází tolik ke křížení vodičů. Více vrstev se také často používá kvůli odstínění jednotlivých vodičů [1]. V takových případech je v sousední vrstvě citlivého nebo rušícího vodiče vedena stínící plocha.

Nakonec jsem zvolil 2 vrstvy spojů. To znamená, že spoje budou z obou stran desky, ale uvnitř žádné vrstvy nebudou. Vodivé propojení mezi vrstvami se realizuje pomocí prokovených otvorů. Pro navrhované zařízení by spoje pouze na jedné vrstvě znamenaly velkou komplikaci při návrhu. Vodiče se v zapojení často kříží, což by vedlo na enormní počet drátových propojek na straně součástek. Naopak použití více jak 2 vrstev je pro navrhované schéma zbytečné a pouze by prodražilo výrobu.

6.1.3 Výběr výrobní technologie

Ještě před začátkem návrhu tištěného spoje je třeba se zamyslet nad výrobou spoje. Je třeba se rozmyslet jakou technologii bude spoj vyráběn a nejlépe i vybrat jeho výrobce. Každý výrobce pak má v podmínkách výroby omezení parametrů tištěného spoje určené tak, aby byl schopen navržený plošný spoj spolehlivě vyrobit. Výrobce zejména sdělí na jakých základních materiálech je schopen vyrábět a s jakou přesností. Z toho se odvíjí požadavky na minimální tloušťku propojovací stopy a minimální velikost izolační mezery mezi spoji. Často je také určen průměr nejmenšího vrtaného otvoru v desce.

Zejména minimální tloušťka propojovacích cest je také dána navrženým obvodem, respektive očekávanými proudy, které potečou po jednotlivých spojích. Velikost izolačních mezer je také zdola omezena používaným napětím v zařízení.

V navrhovaném zařízení se neočekává větší napětí jak 15 V. Proto velikost izolačních mezer není kritická. Také protékající proudy nebudou ani zdaleka dosahovat hodnot, které by mohly zahřívat tenké spoje. Naprostá většina spojů v zařízení vede digitální signály pro technologii CMOS. Protékající proudy proto budou spíše nárazové, pouze pro nabití kapacity vedení a vývodů. Samotný ustálený proud je pak takřka na hranici měřitelnosti. Největší proud poteče v napájení ve vedeních v okolí pojistky. Navržená pojistka se má přetavit při proudu 0,5 A, ale v běžném provozu očekávám proudy mnohem menší, hrubý odhad je asi 150 mA. Takový proud nevyžaduje speciálně zesílené stopy.

Po zvážení výše uvedených důvodů jsem zvolil minimální velikost izolační mezery 0,25 mm, minimální šířku stopy 0,4mm a minimální výsledný průměr prokoveného otvoru 0,4mm. Tyto hodnoty s velkou rezervou vyhovují jak použitým napětím a proudům, tak i požadavkům výrobců na spolehlivou výrobu spoje.

Dále je potřeba vybrat typ základního materiálu desky a tloušťku základní měděné folie na materiálu. Materiál se vybírá zejména kvůli svému chování při vysokých frekvencích [1]. Navrhované zařízení ale nepoužívá nijak vysoké frekvence, proto v tomto případě na volbě

materiálu příliš nezáleží. Tloušťka folie se volí podle velikosti proudů. Tlustší folie znamená větší průřez mědi pro vedení proudu a tím menší zahřívání. Jak už bylo několikrát zmíněno, navrhované zařízení nepoužívá žádné vysoké proudy. Proto volím naprosto standardní materiál RF4 s měděnou folií 18 μm . Materiál FR4 se používá na široké spektrum zařízení a je běžně dostupný. Folie tloušťky 18 μm je také naprosto běžná a počítá se s jejím nakovením během výroby prokovených otvorů v desce. Výsledná tloušťka měděné folie tak bude asi 30 μm . Plošný spoj s těmito konstrukčními omezeními a materiály je schopná vyrobit naprostá většina výrobců plošných spojů. Výběr obvyklých materiálů také příznivě ovlivňuje cenu výroby.

6.2 Návrh desky plošného spoje

Po výběru pouzder součástek a omezení parametrů návrhu je možné konečně přistoupit k vlastnímu návrhu plošného spoje. Pro postup návrhu plošného spoje neexistuje jednoznačný postup nebo metodika [1]. Kvalita návrhu vždy záleží na zručnosti a zkušenostech autora. V každém případě je dobré se během návrhu držet předem vybrané koncepce.

Základní koncepci návrhu jsem zvolil následující. Zařízení se bude skládat z jediné desky plošného spoje, nad kterou bude připevněn displej. Vzhledem k velikosti schématu a množství použitých součástek očekávám výslednou velikost desky srovnatelnou s velikostí displeje. Proto displej bude umístěn nad deskou. Připojen bude pomocí konektoru tak, aby bylo možné ho v případě potřeby demontovat.

6.2.1 Pravidla pro umístění součástek

Jednotlivé součástky je pak třeba rozdělit do 2 skupin. Do první skupiny spadají součástky, které musí být vidět, nebo musí být přístupné. To jsou součástky, které musí být umístěny na přední straně desky mimo obrys displeje tak, aby byly snadno dostupné. Do této skupiny spadají všechny svítivé diody, tlačítka a konektory.

Do druhé skupiny pak spadají ostatní součástky, které nemusí být přístupné ani viditelné. Ty mohou být umístěny kdekoliv, i na opačné straně desky.

Návrh tedy začíná umístěním displeje doprostřed plochy. Tím je daná pozice konektoru pro připojení displeje a pozice vrtaných otvorů pro mechanické připevnění sloupků displeje. Po umístění displeje je třeba rozmístit ovládací tlačítka. Ta se umísťují po dvojicích na pravou i levou stranu displeje. Následuje umístění konektoru CON4 pro připojení kabelu k vozidlu. Ostatní konektory se umísťují, až když přijde řada na návrh jejich spojů. Umístění konektorů není kritické, ale platí pravidlo, že se konektory umísťují na okraje výsledné desky. Rozhodně nesmí být žádný konektor umístěn pod displejem.

Následně se již může začít s umísťováním ostatních součástek a postupnému tvoření obrazu spojů pro obě vrstvy. Součástky i spoje je možné umísťovat na obě strany desky, ale je při tom třeba dbát na minimalizaci počtu potřebných prokovených otvorů. Velké množství otvorů v desce může způsobit kroucení desky vlivem vnitřního napětí a také zmenšit mechanickou odolnost na ohyb. Při umísťování součástek je třeba dbát na co nejkratší spoje mezi nimi. Další důležitý aspekt při umístění a natočení součástky je počet nutných křížení vodičů. Křížení vodičů se musí řešit přesunem spoje na opačnou stranu desky, což zvyšuje počet nutných prokovených otvorů.

Při návrhu je vhodné vytipovat skupiny součástek, které tvoří uzavřený celek. Pak je možné se pro celý celek rozhodnout na kterou stranu desky bude umístěn.

Obecně je vhodné dodržovat krátké vodiče mezi součástkami, ale ne vždy je to možné. Nicméně umístění kondenzátorů je třeba věnovat zvláštní pozornost. V navrhovaném zařízení se vyskytují kondenzátory hlavně jako filtrační prvky na napájecím vedení. Takové kondenzátory se vždy funkčně vztahují ke konkrétní součástce. Kondenzátor pak musí být umístěn co nejblíže napájecích pinů součástky. Tomuto pravidlu je třeba podříditi i umístění ostatních součástek. Kondenzátor vzdálený více jak asi 1 cm od napájecích vývodů součástky již nemůže plnit svoji filtrační funkci a stává se zbytečným [1]. Umístění filtračního kondenzátoru daleko od související součástky se považuje za hrubou začátečnickou chybu při návrhu plošného spoje.

6.2.2 Vedení spojů

Na dvouvrstvém plošném spoji se pro vedení signálů používají obě vrstvy. Nicméně při vedení napájení je situace jiná. Během návrhu jsem se snažil dodržet koncepci kdy se rozvody +5 V vedou hlavně po přední straně desky, zatímco rozvod země je veden hlavně po zadní vrstvě. To umožňuje po dokončení návrhu provést tzv. rozlití mědi. Tento termín znamená, že se na každé vrstvě vybere jeden spoj, jehož stopy se rozšíří na celou plochu tak, aby ještě neporušily minimální izolační mezery od ostatních vodivých ploch. Tímto způsobem se vyplní nevyužité oblasti plošného spoje. U rozvodů napájení je největší pravděpodobnost, že tato síť bude nejdelší a rozprostřená po co největší ploše desky. Proto se k rozlití vybírají obvykle napájecí vedení.

Právě rozlévání je důvodem proč při návrhu umísťuji kladný pól napájecího vedení hlavně na jednu stranu desky a záporný pól na druhou stranu desky. Velké plochy měděné folie na výsledné desce mají hned několik účelů.

Prvním účelem je rozvádění tepla a tím chlazení součástek [1]. Měděná folie rozvádí teplo mnohem lépe než laminát základního materiálu. Druhým účelem je alespoň částečné stínění důležitých součástek před indukcí z vnějšího prostředí. Stejně tak plochy rozlité mědi významně omezují intenzitu elektromagnetického vyzařování z desky zařízení [1].

Třetím účelem je omezení plochy proudových smyček, v tomto případě na napájecím vedení. Je-li napájení tvořeno souvislými vrstvami mědi, pak proud k součástce a zpět teče nejkratší možnou cestou. Tím se zmenšuje celková plocha smyčky, kterou musí proud protéct. To je důležité zejména kvůli vyzařování rušení do okolí [1]. U proudové smyčky je třeba si uvědomit, že se jedná vlastně o jeden závit cívky. Proud tekoucí takovou smyčkou vyvolává magnetický tok, jehož tvar a intenzita jsou závislé na ploše smyčky. V ideálním případě poteče zpětný proud v opačném směru přesně pod trasou přitékajícího proudu. V takovém případě se magnetická pole vyvolaná jednotlivými směry proudů vzájemně ruší, čímž dojde k potlačení rušivého vyzařování ze zařízení.

S vyzařováním souvisí také další požadavek na vedení signálů na desce. V blízkosti součástek citlivých na vnější rušení se neumísťují vedení na kterých lze předpokládat signál o vyšších kmitočtech. To platí hlavně pro vedení komunikačních linek. V oblastech umístění taktovacích krystalů se pak pro jistotu nevedou žádné spoje, samozřejmě s výjimkou připojení samotných krystalů. Na opačnou stranu desky pod krystalem a jeho přívody k procesoru se vždy umísťuje plná oblast mědi, která je uzemněna. Tato oblast slouží jako stínění před rušením krystalu a jeho oscilátoru. Toto je zejména důležité v oblasti krystalu pro obvod reálného času. Jakákoliv indukce rušení na vedení krystalu může ovlivnit stabilitu frekvence krystalu a tím zhoršit přesnost měření času v obvodu. Proto se na navržené desce vedené spoje pečlivě vyhýbají oblastem umístění všech krystalů a to na obou vrstvách.

6.2.3 Umístění servisního potisku

Servisní potisk sice nemá přímo vliv na funkčnost zařízení, ale může výrazně zjednodušit osazování plošného spoje. Servisní potisk by měl znázorňovat umístění součástky na desce. Pokud musí být součástka správně orientována, například kvůli polaritě napájení, musí potisk obsahovat i informaci o orientaci součástky. Vhodný program pro tvorbu návrhů desek plošných spojů umisťuje servisní potisk společně se součástkou, návrhář tedy má zjednodušenou práci.

Potisk obsahuje i označení součástky, obvykle označení ze schématu. Tyto popisy je třeba umístit tak, aby bylo jednoznačné. Zejména v oblastech, kde jsou součástky umísťovány blízko sebe se musí jejich popis umísťovat tak, aby bylo jasné ke které součástce patří. Zatímco obrys součástky a polarita může zůstat po osazení součástky skrytá, tak označení součástky musí zůstat viditelné i po osazení všech součástek.

Je třeba dávat pozor, aby servisní potisk nezasahoval do pájecích plošek pro součástky. Pokud by se tak stalo, pak v lepším případě popis nebude čitelný. V horším případě barva na pájecí plošce způsobí špatné zapájení součástky [1].

Do servisního potisku lze také umístit další dodatečné texty. Proto jsem do potisku umístit i význam jednotlivých svítivých diod a konektorů. Poté není třeba hledat význam konektoru nebo diody ve schématu podle označení součástky. Doporučuji takové popisy umístit zejména v případě, kdy je na desce osazeno více stejných konektorů, ovšem s rozdílným zapojením. To je i případ navrhované desky, kdy ISP i JTAG konektor mají stejné provedení. Proto jsou také popsány v potisku na desce.

6.2.4 Data pro výrobu desky plošných spojů

Výroba oboustranných desek plošných spojů s prokovenými otvory je poměrně náročná na technologické vybavení. Výroba navržené desky amatérským způsobem je velmi nevhodná. Je proto lepší samotnou výrobu svěřit specializované firmě. Kvalitní výroba plošných spojů však vyžaduje také kvalitní přípravu podkladů. Na datovém formátu pro předání výrobních dat, je třeba se vždy dohodnout s konkrétním výrobcem.

S přípravou výrobních dat obvykle pomůže zvolený návrhový software. Pro zadání výroby oboustranného plošného spoje je třeba připravit minimálně: Masky měděných obrazců pro obě strany, nepájivé masky pro obě strany a program pro souřadnicovou vrtačku. Volitelně pak i potisk obou stran.

Kapitola 7

Firmware zařízení

Navrhované zařízení se skládá z hlavní desky s displejem a procesorem a z desky ovládacího modulu na volant. Obě desky obsahují vlastní procesor, je tedy třeba vyvinout 2 různé programy. Jeden pro obsluhu ovládacího modulu a druhý pro procesor hlavní desky. Tyto 2 programy se velmi liší z hlediska množství vykonávané práce. Proto k jejich vývoji je třeba přistupovat odlišně. Nejprve bude popsán firmware pro ovládací modul. Až poté bude popsán firmware hlavní desky a přístup k jeho vývoji. V rámci popisu firmware hlavní desky je také zběžně popsán operační systém FreeRTOS použitý na hlavní desce. Nicméně před vlastním vývojem je třeba vybrat programovací jazyk pro vývoj.

7.1 Výběr programovacího jazyka

Obecně při vývoji vestavěných zařízení se obvykle používají velmi jednoduché procesory s velmi omezenou kapacitou programové a operační paměti. U těchto procesorů tak obvykle nelze použít vysokoúrovňové jazyky s automatickou správou paměti jako je třeba Java nebo .net. V případě použitých procesorů řady ATmega je výběr poněkud ulehčen. Existují 2 možnosti.

Procesory lze samozřejmě programovat pomocí assembleru, což je obvykle příliš náročné. Druhou možností je použít programovací jazyk C. Výrobce procesorů, firma Atmel, dodává k procesorům zdarma překladač z jazyka C do assembleru a strojového kódu procesorů. Překladač je založen na oblíbeném překladači GCC, který je upraven pro generování kódu pro procesory Atmel. V tomto případě je volba programovacího jazyka jasná. Jazyk C umožňuje dostatečnou úroveň abstrakce, ale přitom stále umožňuje přístup k jednotlivým registrům a ovládání celého procesoru.

Překladač GCC nabízí také propracované možnosti optimalizací při generování kódu. V oblasti jednoduchých procesorů pro vestavěné aplikace jsou optimalizace kódu důležité. Paměť pro program i pro data je obvykle velmi omezená.

7.2 Ovládací modul

Ovládací modul je velmi jednoduchý. Obsahuje pouze 4 tlačítka a komunikační linku I2C na hlavní desku. Firmware je proto také velmi jednoduchý. Celá obsluha se provádí v hlavní smyčce programu. Přerušování nejsou používána.

Hlavním úkolem programu je ošetření zámků tlačítek a odeslání události o stisku tlačítka procesoru na hlavní desce. Problém je v zámkách tlačítek. Kontakt v tlačítku se

přepíná velmi rychle. Proto při stisku tlačítka do sebe kontakty narazí tak rychle, že dojde ke krátkému odskočení kontaktů od sebe a opět k jejich spojení. Takový odskok se po stisku tlačítka může opakovat i několikrát.

Pokud by zákmity tlačítek nebyly filtrovány v programu, docházelo by při stisku tlačítka ke generování více událostí stisku. Proto se zjišťují stavy tlačítek v programu pouze jednou za 20 ms. Zjištěný stav tlačítka se pak předá funkci *ProcessButton*. Funkce provádí kontrolu stavu a stisknutí tlačítka oznámí pouze jednou na jeden stisk. Filtrace se provádí tak, že tlačítko je považované za stisknuté jen tehdy, pokud byl stisknutý stav zjištěn dvakrát po sobě.

Pokud je třeba oznámit událost stisku tlačítka, je spuštěna funkce *ReportButtonPress*. Funkce zajistí odeslání odpovídajícího kódu klávesy pomocí sběrnice I2C do procesoru hlavní desky. Funkce pouze ovládá řídicí registry vestavěného modulu pro komunikaci na I2C sběrnici. Na dokončení přenosu se čeká aktivně. Během čekání na dokončení přenosu modul nemůže dělat nic jiného a to ani snímat nové stisky kláves. Nicméně odeslání je natolik rychlé, že uživatel není schopen tak rychle mačkat jednotlivá tlačítka aby tuto prodlevu mohl pozorovat.

To je celá činnost firmware ovládacího modulu. Na desce jsou i připravené 3 diody napojené na PWM kanály, nicméně pro diody zatím není žádné využití. V programu jsou sice připravené funkce pro ovládání jasu diod, nicméně se zatím nepoužívají.

Komunikace po sběrnici I2C probíhá jednosměrně. Ovládací modul se chová jako master na sběrnici a vysílá vždy jeden bajt na adresu procesoru hlavní desky. Ten se chová jako slave na sběrnici. Adresa procesoru hlavní desky je určena na číslo 10 v desítkovém zápisu. Toto uspořádání umožňuje, aby bylo k hlavní desce připojeno více ovladačů. Pokud budou obsazovat sběrnici správně, s ohledem na fakt, že sběrnice pak může mít více master uzlů, pak mohou všichni odeslat data do hlavní desky. Aktuálně je vyvinut jeden ovládací modul, takže není třeba řešit problémy více master uzlů.

7.3 Operační systém FreeRTOS

Již nyní je jasné, že firmware hlavní desky bude vykonávat několik vzájemně nezávislých činností. Například odesílat data pro displej a přitom snímat stav tlačítek uživatelského ovládání. Nicméně použitý procesor je jenom jeden a má jenom jedno výpočetní jádro. Je tedy schopen v každém okamžiku vykonávat pouze jednu činnost. Pokud je nezávislých činností málo, lze to řešit ve firmware tak, že se v průběhu déletrvajících funkcí provádí odskok do kontrolní funkce, která rychle zjistí například stavy tlačítek a nutnost reakce si poznačí do paměti k pozdějšímu zpracování. Takový přístup vede na velmi nepřehledný kód, který je náchylný k chybám a špatně se udržuje.

Běžné osobní počítače také vykonávají více nezávislých programů než je počet procesorových jader. funkce je zajištěna tak, že se jednotlivé programy v běhu na procesoru rychle střídají, čímž se vytváří dojem paralelního běhu. Právě střídání jednotlivých procesů a rozhodování který proces teď poběží na procesoru je úlohou jádra operačního systému. Systém FreeRTOS je právě takovým operačním systémem. Tento systém je oproti operačním systémům známým z osobních počítačů velmi jednoduchý. Poskytuje pouze služby plánovače procesů. Díky tomu má dostatečně malý a úsporný kód. To předurčuje použití právě v podobných mikroprocesorech, jako je ten v navrhovaném zařízení. Systém FreeRTOS je dostupný zdarma ke stažení pod licencí GPL [12].

Mimo plánovače systém obsahuje ještě implementaci různých synchronizačních prostředků pro procesy, jako jsou semaforey, mutexy a fronty. Všechny tyto prostředky po-

skytují s podporou operačního systému pasivní čekání procesu. To znamená, že pokud proces nemůže pokračovat v běhu, například kvůli zablokování na mutexu, operační systém mu nebude přidělovat procesorový čas a proces tak nebude provádět zbytečné výpočty a zdržovat tak ostatní procesy. Jakmile důvod blokace pomine, proces bude opět plánován k běhu podle své priority.

7.3.1 Přepnutí kontextu

Jak vlastně probíhá přepínání běžících úloh na procesoru? Jedná se o službu operačního systému. K přepnutí běžící úlohy může dojít v důsledku 2 typů událostí. Běžící úloha se může vzdát dalšího běhu dobrovolně. Například v situaci, kdy nejsou dostupná další data ke zpracování, nebo je naplánováno zastavení úlohy na určenou dobu. Pak se dostává ke slovu jádro operačního systému.

Druhou možností je přerušování běhu úlohy z rozhodnutí operačního systému. K tomu dochází obvykle v pravidelných intervalech. Nastavení intervalu je nezávislé na běžící úloze a ta nemůže ovlivnit, kdy dojde k přerušování. K přerušování běhu dochází proto, aby se při běhu na procesoru mohly střídat různé úlohy, přičemž všechny úlohy jsou připraveny k běhu a tedy se nechtějí vzdát procesorového času dobrovolně.

Jakmile se dostane ke slovu jádro operačního systému, musí uložit aktuální stav běhu přerušované úlohy. Přesný postup se může na jednotlivých platformách lišit. Na platformě ATmega to znamená uložit obsahy všech registrů na zásobník úlohy. Uložen je i ukazatel na následující instrukci. Tím dojde k uchování stavu procesu po dobu, kde běží jiný proces.

Po uložení stavu plánovač operačního systému vybere další proces, který je připraven k běhu. Stav vybraného procesu je obnoven ze zásobníku a proces je spuštěn přesně v takovém stavu, v jakém byl naposledy přerušován.

Každý proces má přiřazenou prioritu běhu. Pro plánovač úloh to znamená, že procesy s větší prioritou mají přednost. Plánovač vždy najde procesy s nejvyšší prioritou v systému, které jsou připraveny k běhu. Tyto procesy pak běží tak dlouho, dokud nejsou zablokovány. Procesy se stejnou prioritou pak dostávají přibližně stejný procesorový čas. S prioritou procesu je tedy nutné pracovat opatrně. Jeden prioritní proces s časově náročným zpracováním může zablokovat ostatní procesy na dlouhou dobu. Prioritu procesům je tedy třeba přiřazovat citlivě, podle důležitosti jejich činnosti. Obecně platí, že vysoce prioritní procesy by měly vykonávat jen časově co nejkratší a nejnutnější zpracování. Po zpracování nejnutnější reakce se musí zablokovat, aby tak umožnily běh dalších, méně prioritních procesů.

Tento přístup také znamená, že proces nemůže ovlivnit okamžik, kdy dojde k přepnutí kontextu na jiný proces. Na jednu stranu se při vývoji kódu jednoho procesu nemusí programátor ohlížet na jiné procesy. Na druhou stranu, pokud různé procesy přistupují ke stejným datům, je třeba přístup synchronizovat, aby nedošlo k porušení konzistence dat vlivem nevhodného proložení běhu procesů. Tím se někdy vývoj kódu procesu opět komplikuje. Chyby v synchronizaci se také obvykle objevují náhodně a nelze je jednoduše nasimulovat. Proto se chyby v synchronizaci procesů obtížně hledají. Zejména s omezenými prostředky pro ladění aplikací vestavěných systémů, může být odhalování synchronizačních problémů velmi náročné.

7.3.2 Úprava pro konkrétní procesor

Systém FreeRTOS je dodáván ve formě zdrojového kódu v jazyce C. Pro konkrétní procesor musí být přeložen a hlavně upraven. Systém je však navržen na snadnost úpravy. Vše je

rozděleno na jádro operačního systému, které je nezávislé na platformě, a na obslužné funkce specifické pro snadnou platformu nebo procesor.

Při návrhu struktury systému byl evidentně kladen důraz na snadné přizpůsobení pro různé platformy. Distribuce systému dokonce již obsahuje velké množství již připravených úprav pro různé platformy. Distribuce obsahuje i úpravy pro použití na procesorech Atmel ATmega. Nicméně tato úprava je realizována pro jiný model procesoru, než je ATmega 1281 osazený v zařízení. Osazený procesor má jinak pojmenované registry pro ovládání vestavěného čítače pro periodické generování přerušení.

Úprava je také určena pro starší typ překladače pro procesory Atmel. Nyní se používá již inovovaný překladač, který bohužel pracuje jiným způsobem s definicí přerušení. Z toho důvodu nelze použít přímo výchozí verzi pro procesory ATmega, ale musel jsem během vývoje sám upravit funkce operačního systému tak, aby byl systém funkční na použitém procesoru.

Naštěstí je většina práce s úpravou již hotová v oficiální distribuci. Zejména je správně implementováno uložení a obnovení stavu procesu. Zbývá tak pouze upravit nastavení čítače a jeho komparátoru ke generování pravidelného přerušení. Ke generování přerušení je použitý čítač č. 0 a jeho výstupní komparátor A. Také se musí upravit název vektoru přerušení pro přepnutí kontextu. Tyto úpravy lze najít ve výsledném kódu v souboru *FreeRTOS/portable/GCC/ATmega168/port.c*. Takto provedená úprava by měla fungovat s nejnovějšími překladači na většině procesorů z řady ATmega.

Na úpravě je také vidět, že samotný systém FreeRTOS zabere z vestavěných modulů pouze jeden čítač a jeho jeden výstupní komparátor. Zbytek vestavěných modulů je ponechán plně k využití programátorovi aplikace.

K použití systému tedy stačí pouze přidat jeho soubory včetně úpravy pro konkrétní platformu k překladu zdrojových kódů. Jádro operačního systému i aplikace se zkompilují a linker je při překladu spojí do jediného programu, který může být nahrán na procesor jako jakákoliv jiná aplikace.

7.3.3 Nastavení systému

Systém FreeRTOS je nutné nastavit pro konkrétní aplikaci. K tomu slouží soubor *FreeRTOSConfig.h*. V souboru se formou preprocesorových maker definují jednotlivá nastavení. Nejdůležitější jsou nastavení nazvaná *configCPU_CLOCK_HZ* a *configTICK_RATE_HZ*. Tato nastavení společně definují interval přerušení pro přepnutí kontextu. Je to tedy nejdelší doba pro nepřerušovaný běh procesu. Z jejich hodnot se vypočítává nastavení čítače generujícího přerušení.

Frekvenci generování přerušení je třeba zvolit vždy podle charakteru vyvíjené aplikace. Velmi krátké časy mezi jednotlivými přerušeními znamenají, že operační systém bude často provádět přepnutí kontextu a výběr dalšího běžícího procesu. Tato činnost nějakou dobu trvá a tím snižuje čas, který by mohl být věnován běhu jednotlivých procesů. Naopak dlouhý čas přerušení znamená dlouhé intervaly na prostrídání jednotlivých procesů. Protože je interval mezi přerušeními i základní jednotkou měření času v operačním systému, dlouhé časy mohou způsobit horší rozlišení času a tím pomalejší reakce a události. Nastavení frekvence přerušení musí vždy vycházet z časových požadavků konkrétní aplikace.

Já jsem pro vývoj firmware zařízení zvolil frekvenci 500 Hz, což znamená základní rozlišení času 2 ms. Toto nastavení tvoří kompromis mezi využitelným výpočetním výkonem a rychlostí reakcí aplikace.

Další nastavení v souboru pak obsahují zejména nastavení dostupnosti jednotlivých

možností a funkcí systému. Popis podrobnějších voleb je již mimo rozsah této publikace. Podrobný popis lze nalézt v dokumentaci systému FreeRTOS.

7.3.4 Základní použití

Použití operačního systému z hlediska programátora vestavěného systému její nijak složité. Jakmile jsou soubory systému přidány k překladu aplikace, je možné používat jednotlivé dostupné funkce. Na začátku je třeba vytvořit funkci *main* jako vstupní bod aplikace. Ve funkci se provede nejdříve nezbytné nastavení registrů procesoru a pak už se pomocí funkce *xTaskCreate* vytvoří jednotlivé procesy a naplánují se ke spuštění pomocí plánovače procesů. Ve funkci se jako parametr jednoduše předá ukazatel na funkci, která implementuje tělo daného procesu.

Tělo procesu musí být vždy implementováno jako nekonečná smyčka. Kód vykonávající funkci tedy nesmí nikdy opustit funkci. Jiné podmínky na kód procesu nejsou a programátor může využívat libovolné prostředky procesoru. Samozřejmě je třeba dávat pozor na synchronizaci procesů při přístupu ke stejným datům.

Po naplánování jednotlivých procesů ještě nejsou procesy spuštěny. Start operačního systému se provede pomocí spuštění funkce *vTaskStartScheduler*. Tato funkce spustí plánovač úloh a nastaví pravidelné generování přerušování. Funkce je navržena tak, že nikdy nevrátí řízení zpět do funkce *main*. Funguje tak jako hlavní smyčka programu. Po spuštění funkce už běží pouze naplánované úlohy, které se na procesoru střídají.

Operační systém nebrání programátorovi ve využívání ostatních přerušování poskytovaných procesorem. Je třeba si však uvědomit, že tělo přerušování se vždy vykonává v kontextu právě běžícího procesu. Nedá se tak ovlivnit ve kterém procesu bude přerušování spuštěno. Z toho důvodu je třeba používat pouze ty funkce operačního systému, které jsou připraveny pro volání z obsluhy přerušování. Takové funkce mají na konci názvu vždy zkratku ISR. Pro psaní obsluhy přerušování platí podobná pravidla jako v případě, že není využitý žádný operační systém. Obsluha přerušování musí být co nejkratší a nesmí zablokovat proces, ve kterém se spustila. Proto funkce operačního systému volané z obsluhy přerušování nikdy neblokují aktuálně spuštěný proces, protože ten nesouvisí s obsluhou přerušování. Místo toho funkce vrací zpět informaci, zda provedením funkce byl připraven k běhu nějaký proces s větší prioritou než má aktuální proces. Pokud ano, musí obsluha přerušování vyvolat funkci pro dobrovolné uvolnění procesoru běžícím procesem. Tím se dostane ke slovu opět operační systém, který provede přepnutí kontextu na prioritnější proces.

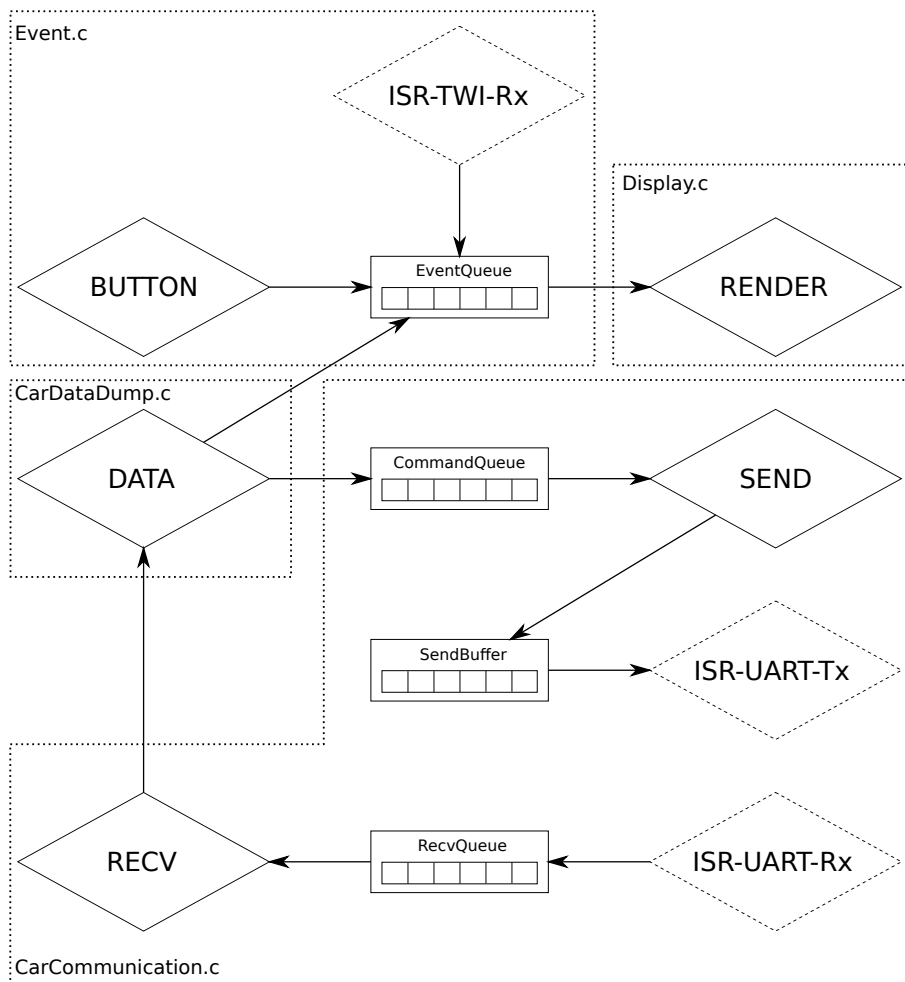
7.4 Firmware pro hlavní desku

Jak už bylo zmíněno, firmware pro hlavní desku používá operační systém FreeRTOS. Firmware je rozdělen na jednotlivé úlohy. Uživatelských úloh je celkem 5, k tomu se přidává obsluha 3 přerušovacích vektorů. Jednotlivé úlohy mezi sebou komunikují a předávají si postupně data ke zpracování. Pro účely výměny dat je mezi procesy několik vyrovnávacích pamětí pro předání dat. Jedna z vyrovnávacích pamětí je tvořena pouze staticky alokovaným polem, ostatní paměti jsou ve formě fronty s různou kapacitou.

Vazby mezi jednotlivými úlohami a směry předávání dat zachycuje obrázek 7.1. Jednotlivé úlohy jsou značeny pomocí kosočtverce s plným obrysem. Pojmenování úloh pro lepší orientaci odpovídá pojmenování v programu. Kosočtverce s čárkovaným obrysem znázorňují obsluhy přerušování. Obsluhy přerušování běží sice samostatně, ale nemají vlastní úlohu v operačním systému. Při vyvolání přerušování se obsluhy vždy nastartují v kontextu právě

běžícího procesu. Šipky znázorňují směr předávání dat mezi procesy. Data se obvykle přehodí pomocí nějaké formy vyrovnávací paměti. Vyrovnávací paměti jsou pojmenovány názvem proměnné, která je v programu zastupuje. Nakonec oblasti oddělené čárkovanými čarami znázorňují ve kterém zdrojovém souboru jsou implementované jednotlivé úlohy, respektive deklarovány vyrovnávací paměti.

Nyní vysvětlím postupně jednotlivé části firmware.



Obrázek 7.1: Architektura procesů a komunikace.

7.4.1 Snímání stavu tlačítek

Snímání stavu tlačítek zajišťuje proces *BUTTON*. Snímání je velmi podobné jako v ovládacím modulu popsáném v kapitole 7.2. Proces je nastaven tak, aby zjišťoval stavy tlačítek zhruba jednou za 20 ms. Je třeba opět řešit zákmity tlačítek. Kód pro filtraci zákmitů tlačítek pracuje stejným způsobem jako v ovládacím modulu. Tedy po každém zjištění stavy tlačítka se stav porovná se stavem zjištěným v minulém průchodu. Pokud již v minulém průchodu bylo tlačítko stisknuté a nyní je stále stisknuté, pak se zahlásí událost stisku odpovídajícího tlačítka. Pokud již byl stisk tlačítka ohlášen, další hlášení se negeneruje dokud není tlačítko puštěné a stisknuté znovu.

Je třeba poznamenat, že stisknuté tlačítko se projeví čtením logické 0 ze vstupního pinu. Událost je tedy generována při sestupné hraně signálu, po odstranění rychlých záskmitů.

Hlášení stisku tlačítka se provede pomocí vygenerování události typu *TEvent*. Událost se vloží do fronty událostí pojmenované *EventQueue*. Fronta událostí má maximální kapacitu 20 událostí. Pokud by se nově vygenerovaná událost do fronty již nevešla, pak nedochází k zablokování procesu, ale k zahození události. Zpracování fronty událostí pak má na starosti jiný proces.

7.4.2 Příjem hlášení od ovládacího modulu

Ovládací modul komunikuje s hlavním procesorem pomocí komunikační linky I2C. Během startu procesu pro obsluhu tlačítek je také připraven na příjem vestavěný modul pro komunikaci po sběrnici I2C [2]. Modul je přepnut do režimu slave a nastavena adresa 10 v desítkovém zápisu. Je povoleno vyvolání přerušení při příjmu dat. Právě obsluha přerušení má na starosti zpracování přijatých dat. Obsluha přerušení je v obrázku 7.1 označena jako *ISR-TWI-Rx*.

Při příjmu dat obsluha přerušení vyzvedne přijatá data z registru modulu. Jedná se vždy o jeden bajt. Podle čísla doručeného po sběrnici je pak vygenerována příslušná událost. Událost je následně vložena do fronty událostí, kde čeká na zpracování. Stejně jako u procesu snímání stavu tlačítek je událost zahozena v případě, že je fronta událostí plná. Obsluha přerušení odesláním události do fronty končí.

7.4.3 Komunikace s displejem a uživatelské rozhraní

Tento proces je ze všech nejrozsáhlejší a také má největší nároky na výpočetní výkon. Proces má na starosti hned několik činností. Už z obrázku 7.1 je patrné, že proces zpracovává frontu událostí. Také udržuje aktuální stav uživatelského rozhraní a provádí přepínání stavů v reakci na příchozí události. Dále sestavuje obraz pro displej a tento obraz následně odesílá do displeje. Z důvodu přehlednosti je kód rozdělen do 2 zdrojových souborů. V souboru *DisplayComm.c* je pouze implementace komunikace s displejem. Vše ostatní se nachází v souboru *Display.c*.

Knihovna U8G

Pro generování obrazových dat je použita knihovna U8g. Knihovna je dostupná zdarma jako open source pod licencí BSD. Je navržena speciálně pro malé 8 bitové procesory. Knihovna podporuje velké množství různých displejů a jejich řadičů, zejména modely s komunikačním rozhraním SPI. Bohužel displej, který je osazený v zařízení, podporován není. Je tedy potřeba si dopsat vlastní ovladač displeje pro knihovnu U8g. Technicky knihovna funguje tak, že nejprve sestaví obraz paměti, který má být nahrán do displeje. Po ukončení generování dat jsou tato data předána ovladači displeje, který je zodpovědný za jejich přenos do displeje. Ovladač pro displej je implementován funkcí *DeviceControl* v souboru *DisplayComm.c*. Jedná se o obsluhu událostí z knihovny, přičemž reaguje na události pro inicializaci a odeslání dat. Zbytek událostí se zpracovává pomocí vestavěných obslužných funkcí z knihovny. Odeslání dat do displeje je realizováno pomocí funkcí popsanych v kapitole 7.4.3.

Zajímavostí je, že knihovna obraz sestavuje postupně po jednotlivých paměťových stránkách a tyto stránky se do displeje přenášejí samostatně. To klade větší nároky na výpočetní výkon, neboť se obraz generuje v 8 průchodech místo jediného. Na druhou stranu jsou sestavovaná data menší, čímž se ušetří nezanedbatelné množství paměti RAM v aplikaci.

Pro sestavování obrazu knihovna obsahuje obvyklé funkce pro kreslení základních geometrických tvarů a pro vykreslení textu. Pro text lze používat široký výběr fontů. Nicméně fontů je v knihovně takové množství, že po kompilaci je drtivá většina velikosti programu tvořena fonty. Proto jsem si vybral 2 fonty, které budu využívat. Ostatní fonty jsou odstraněny.

Komunikace s displejem

Jak už bylo řečeno v kapitole 5.2.2, displej je připojen pomocí 20 vodičů. K procesoru je však přivedeno jen 15 z nich. Displej používá paralelní 8 bitové rozhraní na data a k tomu dalších 7 pomocných signálů. Z hlediska komunikace je důležité, že displej obsahuje 2 řadiče. Každý z řadičů obsluhuje polovinu displeje. Každý řadič je však nezávislý a má svůj stavový registr. Řadiče obsahují RAM paměť na obrazová data. Každý řadič obsluhuje plochu 64 krát 64 pixelů. Každý pixel má jenom 2 možné stavy, takže obrazová data obsahují právě jeden bit na jeden pixel.

Paměť v řadiči je organizována do 8 stránek, přičemž každá stránka obsahuje 64 bajtů dat. Z hlediska zobrazení je bajt orientován na výšku, každá stránka tak obsahuje obraz v horizontálním pruhu vysokém 8 pixelů.

Při komunikaci se nejdříve musí vybrat řadič, se kterým se bude komunikovat. Pak se dalším signálem nastaví zda se jedná o řídicí příkaz nebo o data. Nakonec se určí jestli se bude jednat o zápis nebo čtení. Pak už jen stačí nastavit data na datovou sběrnici a poslat impuls na povolovací vstup. Při komunikaci se musí dodržet předepsané časování. Podle výrobce je největší možná frekvence komunikace 1 MHz. Procesor by však byl schopen komunikovat rychleji. Proto je třeba mezi jednotlivými přepnutími vodičů vkládat čekání.

Vkládat čekání realizované operačním systémem není vhodné. Potřebné časy se pohybují na úrovni jednotek us nebo i kratších. Operační systém není schopen pracovat s tak malou časovou konstantou. Vzhledem k taktovací frekvenci je nejjednodušší vložit na potřebná místa instrukce NOP a tím vynutit čekání. Vzhledem k potřebným časům není nikde třeba vkládat více jak 8 prázdných instrukcí. Program komunikace s displejem tedy používá pro dodržení časování aktivní čekání, nicméně časy čekání jsou natolik krátké, že aktivní čekání nijak nevádí a pasivní čekání by se v tomto případě realizovalo obtížně.

Uživatelské rozhraní

Uživatelské rozhraní se skládá z úvodního menu a jednotlivých obrazovek. Stav uživatelského rozhraní je uložen v proměnné *UIStateSelected*. Jedná se o výčtový typ, který označuje právě aktivní obrazovku uživatelského rozhraní.

Rozhraní se obsluhuje pomocí tlačítek rozmístěných okolo displeje, případně pomocí stejně rozmístěných tlačítek na ovládacím modulu. Tlačítka na levé straně displeje provádějí pohyb mezi úrovněmi uživatelského rozhraní. Horní tlačítko znamená vstup do vybrané položky menu, tlačítko vlevo dole pak výstup o úroveň výš. Tlačítka na pravé straně slouží pro pohyb nahoru a dolů.

Úvodní menu se skládá ze 3 položek. První položkou je *Terminal*. Tato obrazovka je to pouze pro ladící účely. Jedná se o seznam posledních sedmi hlášení. V hotovém a odladěném produktu by taková obrazovka nebyla, nicméně zařízení vytvořené v této práci je prototyp a proto jsem obrazovku pro názornost ponechal.

Druhou položkou je *Brightness*. Jedná se o menu k nastavení intenzity podsvícení displeje. Intenzita podsvícení se dá nastavit ve stupních od 0 do 10. Nastavuje se pomocí tlačítek nahoru a dolů.

Třetí položka menu označená *Data from car* je určená k zobrazení údajů nečtených z automobilu.

Udržování informací o hlavním menu má na starosti proměnná *MainMenu*. Ve struktuře je jak definice textů pro jednotlivé položky menu, tak i proměnná s aktuálně vybraným řádkem. Vzhledem ke struktuře menu je velmi jednoduché rozšiřování menu o další položky. Implementace si poradí i s případem, kdy bude položek v menu více, než se vejde na displej. V takovém případě se menu začne zobrazovat po stránkách.

Jakmile má dojít k vykreslení obrazu, je spuštěna funkce *DrawUI*. Funkce pak spustí vykreslování podle toho, jaká obrazovka je právě zvolena. Vykreslení údajů může provádět každá obrazovka po svém.

K vlastnímu vykreslování obrazu je použita grafická knihovna U8g. Stručný popis knihovny najdete v kapitole 7.4.3. Knihovna obsahuje obvyklé funkce pro vykreslení základních geometrických tvarů a vykreslení textu zvoleným fontem. Po sestavení dat pro displej pak dojde k jejich odeslání do displeje.

Nastavení intenzity podsvícení

Podsvícení displeje se ovládá pomocí PWM signálu generovaného pomocí vestavěného čítače č. 0 a jeho výstupního komparátoru A, viz kapitola 5.3.3. Generování signálu je nastaveno na frekvenci přibližně 1 kHz. Pomocí změny poměru doby zapnutí a vypnutí diody se lidskému oku zdá, že se mění intenzita svitu diody. Ve skutečnosti však dioda pouze bliká.

Po nastavení čítače je změna intenzity jednoduchá. Stačí nastavit hodnotu výstupního komparátoru na požadovanou úroveň. Přičemž hodnota 0 na komparátoru znamená zhasnutou diodu a hodnota 255 znamená plné svícení.

Zpracování událostí

Události se zpracovávají po jedné, v pořadí jak jsou uloženy ve frontě událostí. Pro každou událost se spustí funkce *ProcessEvent*. Tato funkce má za úkol pouze předat zpracování události do specifické funkce podle toho, která obrazovka uživatelského rozhraní je právě aktivní. Každá obrazovka si pak provádí své zpracování příchozích událostí.

Zpracovávají se vždy všechny události aktuálně zařazené ve frontě událostí. Jakmile je fronta prázdná, spustí se znovu vykreslování na displej aby se tak vykreslily změny, které mohla událost způsobit. Po překreslení displeje je proces zablokován, dokud není ve frontě připravena další událost.

7.4.4 Získávání dat z vozidla

Proces pro získávání dat z vozidla je na obrázku 7.1, označen názvem *DATA*. Tento proces pracuje s již připravenými službami pro komunikaci, takže získávání dat není nijak komplikované. Proces si zvolí příkaz, který má být odeslán vozidlu. Zároveň si vytvoří pole pro uložení odpovědi od vozidla. Vybraný příkaz a ukazatel na prostor pro odpověď předá funkci *CarCom_Command*. Funkce zajistí naplánování odeslání příkazu a zablokuje proces. Až jsou data nachystána v poli pro odpověď, je proces opět probuzen a může si volně manipulovat s odpovědí. Vytvořené příkazy čekající na odeslání se vkládají do fronty *CommandQueue*. Aktuálně jsou podporované příkazy pro zjištění VIN kódu vozidla a z provozních údajů příkazy pro zjištění aktuální rychlosti, otáček motoru a teploty chladicí kapaliny. Pro ověření funkce komunikace v prototypu takové údaje stačí. Nicméně program je připraven na snadné přidání dalších podporovaných údajů.

Takto se získají jednotlivé definované údaje od automobilu a výsledky se naformátují do textové podoby. V textové podobě jsou pak zobrazovány na obrazovce pro výpis údajů z vozidla.

Po zjištění všech naplánovaných údajů proces vyšle do fronty událostí příkaz k překreslení displeje. Tím se aktualizují údaje na displeji. Údaje z vozidla se zjišťují pravidelně jedenkrát za vteřinu.

7.4.5 Odeslání příkazu pro čip ELM327

Na odeslání příkazu nejprve pracuje proces na obrázku 7.1 označený názvem *SEND*. Ten je zodpovědný za odebírání plánovaných příkazů z fronty *CommandQueue*. Proces vždy odebere jeden příkaz. Pokud ve frontě žádný příkaz není, operační systém proces zablokuje dokud nebude nějaký příkaz k dispozici. Po výběru příkazu se musí počkat, až bude vyřízen předchozí příkaz a komunikační čip bude připraven přijmout další příkaz. K tomu se používá synchronizační semafor *CommandSemaphore*. Ten je odemčen přijímacím procesem po obdržení vyzývacího znaku.

Vyzvednutý příkaz se nejprve převede do textové podoby a vloží se do proměnné *SendBuffer*. Pak se povolí přerušení na odesílání dat. Odesílání dat probíhá po jednotlivých znacích. Proměnná tak slouží jako předávací paměť pro celý řetězec při komunikaci mezi procesem a přerušením pro odesílání dat.

Následně je povoleno přerušení pro odesílání dat, na obrázku 7.1 označené názvem *ISR-UART-Tx*. To je vyvoláno vždy, když je jednotka pro sériovou komunikaci připravena přijmout do registru další bajt k odeslání. Jakmile se odešle potvrzovací znak na konci příkazu, obsluha přerušení zakáže další aktivaci přerušení. Přerušení je později opět povoleno až když je připraven další příkaz.

7.4.6 Příjem odpovědi na zasláný příkaz

Vestavěný modul pro sériovou komunikaci přijímá data po komunikační lince po jednotlivých bajtech. Vždy po příjmu dalšího bajtu je vyvoláno přerušení, které má přijatý bajt vyzvednout z registru dříve, než dojde k jeho přepsání dalším přijatým bajtem. Přerušení je na obrázku 7.1 označené názvem *ISR-UART-Rx*. Obsluha přerušení pouze vyzvedne přijatý bajt z registru a zařadí jej do fronty *RecvQueue*.

Přijaté znaky jsou pak z fronty vybírány procesem pojmenovaným *RECV*. Tento proces provádí filtraci přijatých znaků tak, aby se na výstup neukládaly nepotřebné znaky. Výsledná odpověď se sestavuje přímo v poli pro vrácení výsledku, viz kapitola 7.4.4.

Po přijetí vyzývacího znaku je jasné, že je odpověď kompletní. Proto se přijatý řetězec ukončí pomocí nulového bajtu a dojde k probuzení procesu, který příkaz odeslal. Ten již má ve svém poli kompletní sestavenou odpověď na příkaz. Zároveň dojde k odemčení semaforu *CommandSemaphore*. To dovolí zpracování a odeslání dalšího příkazu.

Kapitola 8

Závěr

Tato práce postupně provádí čtenáře návrhem vestavěného systému, který má fungovat jako palubní počítač v automobilu. Text práce čtenáře provádí postupně všemi kroky návrhu takového zařízení. Jednotlivé kapitoly často upozorňují na rizika, která hrozí v dané fázi návrhu zařízení. Text také zdůvodňuje všechna důležitá rozhodnutí během návrhu a stručně komentuje uvažované možnosti včetně jejich výhod a nevýhod.

V rámci praktické části práce byly všechny kroky také fyzicky provedeny. Součástí výsledků práce tak je vyrobený a funkční prototyp zařízení. Během práce jsem tak provedl osazení a zapájení desek plošných spojů, jejich oživení i vývoj základního firmware. Pouze výroba samotné hlavní desky plošného spoje podle provedeného návrhu byla realizována formou zakázky.

Tato práce byla náročná z hlediska nutnosti zkombinovat znalosti a zkušenosti z oblasti návrhu elektroniky a zároveň z oblasti tvorby software. Přesně tak, jak se od návrhu a realizace vestavěného systému čeká.

Zapojení hardware i struktura software jsou provedeny velmi univerzálně. Je tedy možné zařízení dále vylepšovat. Zejména se nabízí rozšíření hodnot čtených z automobilu. Dále se dají čtené hodnoty rozšířit o čtení diagnostických kódů. Je také možné připojit jiný displej a vylepšit tak uživatelské rozhraní. Uživatelské rozhraní lze také vylepšit inovovaným ovládacím modulem. Pro lepší vzhled zařízení by bylo vhodné navrhnout a vyrobit krabičku. Velmi zajímavé by bylo rozšíření zařízení o možnost záznamu provozních dat, například na SD kartu. Takové rozšíření by však vyžadovalo změnu v hardware.

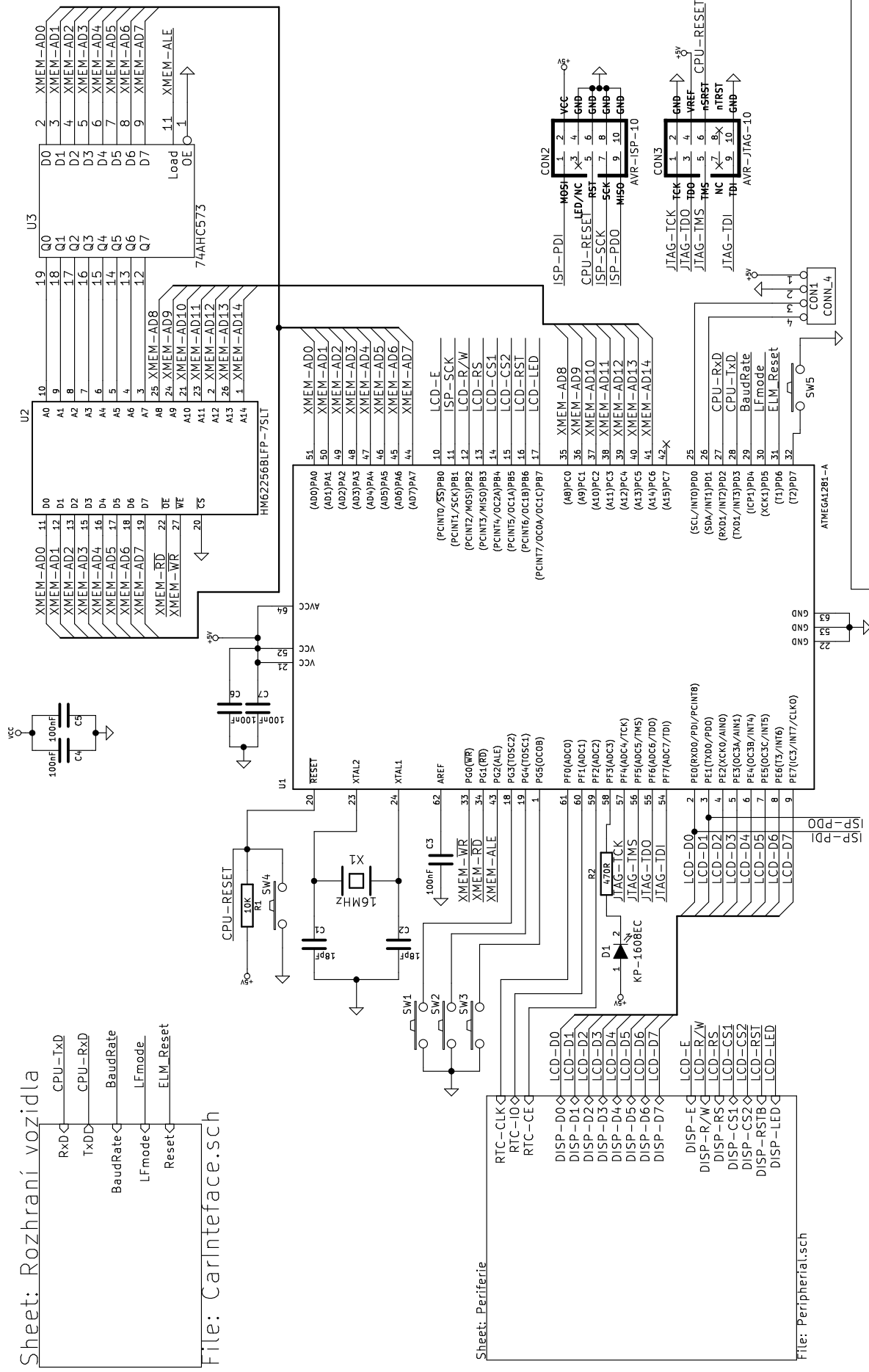
Literatura

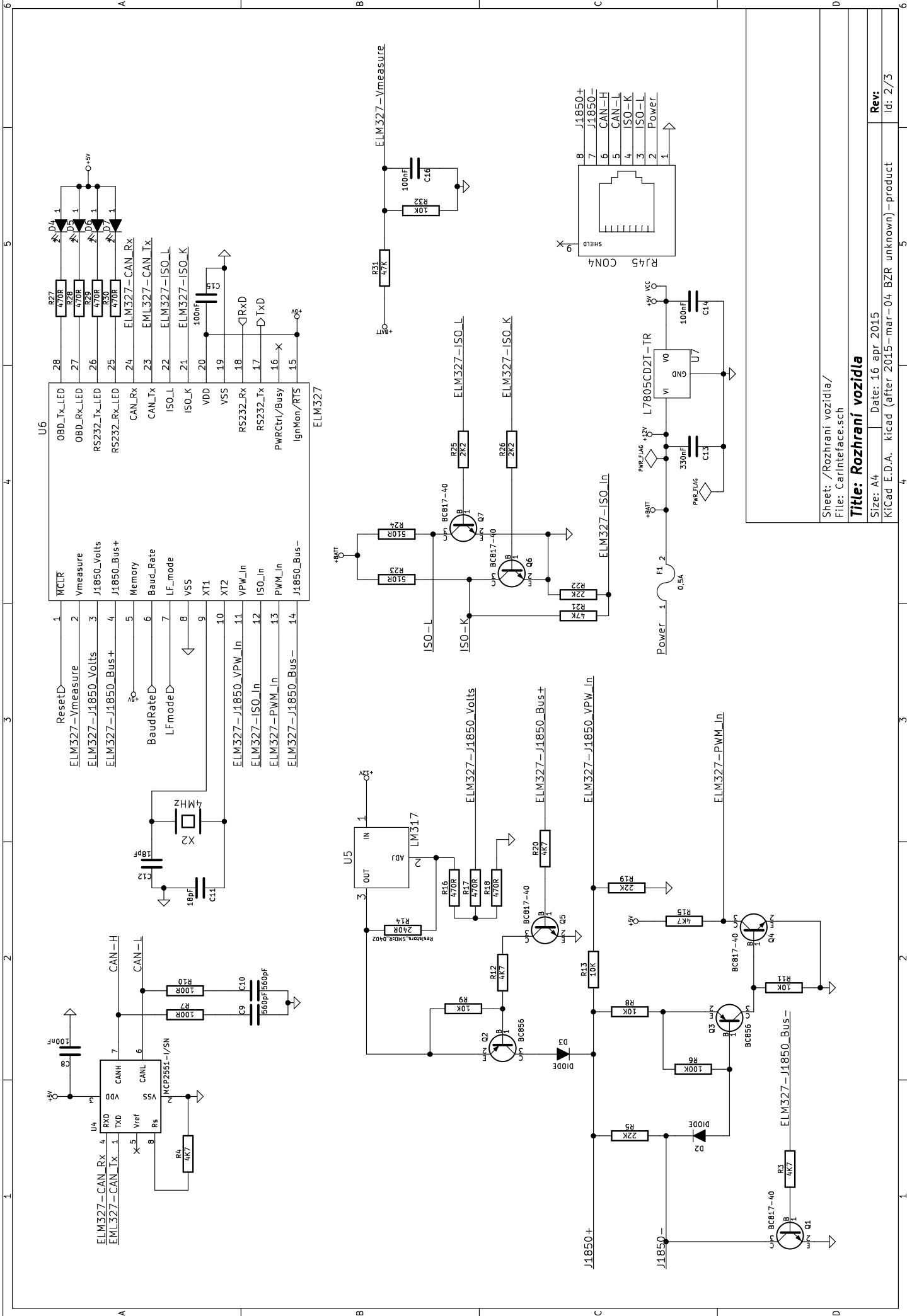
- [1] ŠANDERA, Josef. Návrh plošných spojů pro povrchovou montáž. 1. vyd. Praha: BEN - technická literatura, 2006. 272 s. ISBN 80-7300-181-0.
- [2] Atmel. 8-bit Atmel Microcontroller ATmega1281 Datasheet.
- [3] Atmel. 8-bit Atmel Microcontroller ATmega88PA Datasheet.
- [4] Dallas Semiconductor. DS1302 Trickle-Charge Timekeeping Chip.
- [5] Elm Electronics Inc. ELM327 OBD to RS232 Interpreter. Dostupné z <http://www.elmelectronics.com/DSheets/ELM327DSH.pdf>.
- [6] Fairchild Semiconductor. MC78LXXA fixed voltage regulator datasheet.
- [7] ISO 15031-3. Road vehicles – Communication between vehicle and external equipment for emissions-related diagnostics – Part 3: Diagnostic connector and related electrical circuits, specification and use. Switzerland: ISO/IEC.
- [8] ISO 15031-6. Road vehicles – Communication between vehicle and external equipment for emissions-related diagnostics – Part 6: Diagnostic trouble code definitions. Switzerland: ISO/IEC.
- [9] ISO 15765-4. Road vehicles – Diagnostic communication over Controller Area Network (DoCAN) – Part 4: Requirements for emissions-related systems. Switzerland: ISO/IEC.
- [10] ISO 9141-2. Road vehicles – Diagnostic systems – Part 2: CARB requirements for interchange of digital information. Switzerland: ISO/IEC.
- [11] Midas Displays. Specification: MC128064A6W-FPTLW.
- [12] Real Time Engineers ltd. FreeRTOS [online]. Dostupné z <http://www.freertos.org>.
- [13] Directive 98/69/EC of the European Parliament. Brusel: European parliament, 1998.

Příloha A

Schéma obvodu

Na následujících stranách najdete schémata výsledného obvodu a ovládacího modulu.

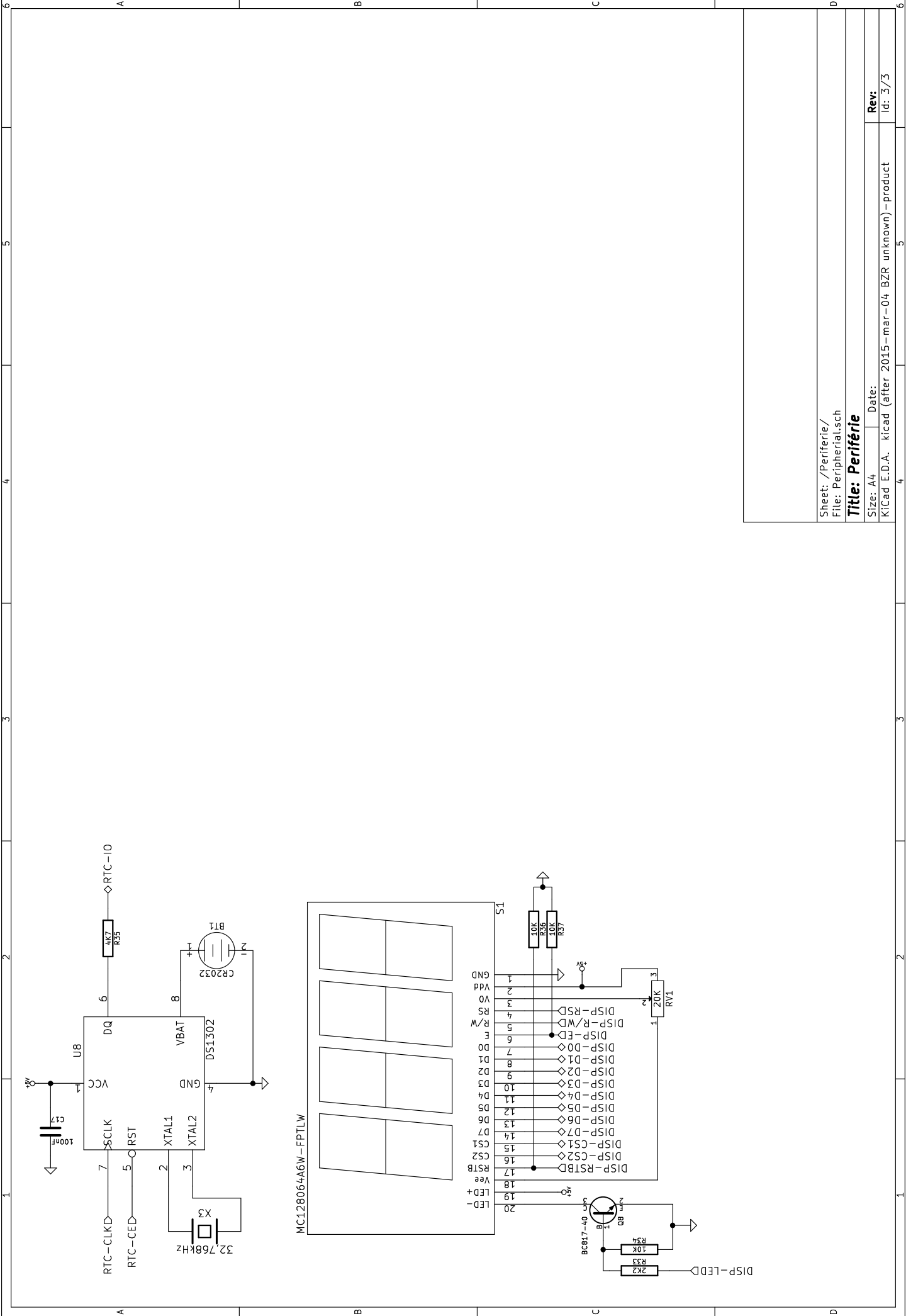




Sheet: /Rozhrani vozidla/
File: CarInteface.sch

Title: Rozhraní vozidla

| | | |
|--|-------------------|---------|
| Size: A4 | Date: 16 apr 2015 | Rev: |
| KiCad E.D.A. kicad (after 2015-mar-04 BZR unknown)-product | | Id: 2/3 |



Sheet: /Periferie/
File: PeripheriaLsch

Title: Periferie

Size: A4 Date:

KiCad E.D.A. kicad (after 2015-mar-04 BZR unknown)-product

Rev:

Id: 3/3

